MMUSIC
Internet-Draft
Intended status: Standards Track
Expires: January 22, 2016

K. Drage, Ed.
M. Makaraju
J. Stoetzer-Bradler
Alcatel-Lucent
R. Ejzak
J. Marcon
Unaffiliated
July 21, 2015

SDP-based Data Channel Negotiation
draft-ietf-mmusic-data-channel-sdpneg-03

Abstract

The Real-Time Communication in WEB-browsers (RTCWeb) working group is charged to provide protocols to support direct interactive rich communications using audio, video, and data between two peers' web-browsers. For the support of data communication, the RTCWeb working group has in particular defined the concept of bi-directional data channels over SCTP, where each data channel might be used to transport other protocols, called sub-protocols. Data channel setup can be done using either the in-band Data Channel Establishment Protocol (DCEP) or using some out-of-band non-DCEP protocol. This document specifies how the SDP offer/answer exchange can be used to achieve such an out-of-band non-DCEP negotiation. Even though data channels are designed for RTCWeb use initially they may be used by other protocols like, but not limited to, the CLUE protocol. This document is intended to be used wherever data channels are used.

Table of Contents

1. Introduction

The RTCWeb working group has defined the concept of bi-directional data channels running on top of SCTP/DTLS. RTCWeb leaves it open for other applications to use data channels and its in-band DCEP or out-of-band non-DCEP protocols for creating them. Each data channel consists of paired SCTP streams sharing the same SCTP Stream Identifier. Data channels are created by endpoint applications through the WebRTC API, or other users of data channel like CLUE, and can be used to transport proprietary or well-defined protocols, which in the latter case can be signaled by the data channel "sub-protocol" parameter, conceptually similar to the WebSocket "sub-protocol". However, apart from the "sub-protocol" value transmitted to the peer, RTCWeb leaves it open how endpoint applications can agree on how to instantiate a given sub-protocol on a data channel, and whether it is signaled in-band using DCEP or out-of-band using a non-DCEP protocol (or both). In particular, the SDP offer generated by the application includes no channel-specific information.

This document defines SDP offer/answer negotiation procedures to establish data channels for transport of well-defined sub-protocols, to enable out-of-band negotiation.

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. Terminology

This document uses the following terms:

Data channel: A WebRTC data channel as specified in [I-D.ietf-rtcweb-data-channel].

Data channel stack: An entity which, upon application request, runs the data channel protocol to keep track of states, sending and receive data. If the application is a browser based JavaScript application then this stack resides in the browser. If the application is a native application then this stack resides in the application and is accessible via some sort of APIs.

Data channel properties: Fixed properties assigned to a data channel at the time of its creation. Some of these properties determine the way the data channel stack transmits data on this channel (e.g., stream identifier, reliability, order of delivery...).

DCEP: Data Channel Establishment Protocol defined in [I-D.ietf-rtcweb-data-protocol].

In-band: Transmission through the peer-to-peer SCTP association.

Out-of-band: Transmission through the application signaling path.

Peer: From the perspective of one of the agents in a session, its peer is the other agent. Specifically, from the perspective of the SDP offerer, the peer is the SDP answerer. From the perspective of the SDP answerer, the peer is the SDP offerer.

SCTP Stream Sequence Number (SSN): the SCTP stream sequence number as specified in [RFC4960].

Stream identifier: The identifier of the outbound and inbound SCTP streams composing a data channel.

4. Applicability Statement

The mechanism in this specification only applies to the Session Description Protocol (SDP) [RFC4566], when used together with the SDP offer/answer mechanism [RFC3264]. Declarative usage of SDP is out of scope of this document, and is thus undefined.

5. Data Channels

This section summarizes how data channels work in general.

A WebRTC application creates a data channel by providing a number of setup parameters (sub-protocol, label, reliability, order of delivery, priority). The application also specifies if it wants to make use of the negotiation using the DCEP [I-D.ietf-rtcweb-data-protocol], or if the application intends to negotiate data channels using the SDP offer/answer protocol.

In any case, the SDP offer generated by the browser is per [I-D.ietf-mmusic-sctp-sdp]. In brief, it contains one "m" line for the SCTP association on top of which data channels will run:

```
m=application 54111 UDP/DTLS/SCTP webrtc-datachannel
c=IN IP4 79.97.215.79
a=max-message-size:100000
a=sctp-port 5000
a=setup:actpass
a=connection:new
a=fingerprint:SHA-1 \
    4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
```

Note: A WebRTC browser will only use "m" line format "webrtc-datachannel", and will not use other formats in the "m" line for other protocols such as t38. [I-D.ietf-mmusic-sctp-sdp] supports only one SCTP association to be established on top of a DTLS session.

Note: This SDP syntax does not contain any channel-specific information.

5.1. Stream Identifier Numbering

[Editor's note: This entire Section 5.1 could possibly be removed as there is another section (Section 6.2.1) which is dedicated to SCTP stream id usage.]

Independently from the requested type of negotiation, the application creating a data channel can either pass to the browser the stream identifier to assign to the data channel or else let the browser pick one identifier from the ones unused.

To avoid glare situations, each endpoint can moreover own an exclusive set of stream identifiers, in which case an endpoint can only create a data channel with a stream identifier it owns.

Which set of stream identifiers is owned by which endpoint is determined by convention or other means.

For data channels negotiated with the DCEP, one endpoint owns by convention the even stream identifiers, whereas the other owns the odd stream identifiers, as defined in [I-D.ietf-rtcweb-data-protocol].

For data channels negotiated via some non-DCEP protocol, no convention is defined by default.

5.2. Generic Non-DCEP Negotiation

[Editor's note: As this document now focuses on SDP offer/answer negotiation only, should this entire Section 5.2 (and all its sub-sections) be removed? Parts of the information provided in these sub-sections might be moved to Section 6, Section 6.2.1 and Section 6.2.3 as indicated below in further editor notes. Another option might be to move this section and its sub-section to an informative appendix.]

5.2.1. Overview

DCEP negotiation only provides for negotiation of data channel transport parameters and does not provide for negotiation of sub-protocol specific parameters. Non-DCEP negotiation can be defined to allow negotiation of parameters beyond those handled by DCEP negotiation, e.g., parameters specific to the sub-protocol instantiated on a particular data channel. See Section 6.1.2 for an example of such a parameter.

[Editor's note: The information in the preceding paragraph might be moved to Section 6.]

The following procedures are common to all methods of non-DCEP negotiation, whether in-band (communicated using proprietary means on an already established data channel) or out-of-band (using SDP offer/answer or some other protocol associated with the signaling channel).

[Editor's note: The preceding paragraph could be deleted.]

5.2.2. Opening a Data Channel

In the case of non-DCEP negotiation, the endpoint application has the option to fully control the stream identifier assignments. However these assignments have to coexist with the assignments controlled by the data channel stack for the DCEP negotiated data channels (if

any). It is the responsibility of the application to ensure consistent assignment of stream identifiers.

[Editor's note: The information in the preceding paragraph could be moved to Section 6.2.1.]

When the application requests the creation of a new data channel to be set up via non-DCEP negotiation, the data channel stack creates the data channel locally without sending any DATA_CHANNEL_OPEN message in-band [Editor's note: The information in this first part of the sentence could be moved to section 6.2.3], and sets the data channel state to Connecting if the SCTP association is not yet established, or sets the data channel state to Open if the SCTP association is already established. The side which starts non-DCEP negotiation creates a data channel using underlying data channel stack API and the data channel is put into open state immediately (assuming ICE, SCTP procedures were already done). [Editor's note: The API related information could be deleted.] However, the application can't send data on this data channel until non-DCEP negotiation is complete with the peer. This is because the peer needs to be aware and accept the data channel via non-DCEP negotiation. The peer after accepting the data channel offer can start sending data immediately. This implies that the offerer may get data channel sub-protocol messages before non-DCEP negotiation is complete and the application should be ready to handle it. [Editor's note: The information in these last four sentences could be moved to Section 6.2.3.]

If the peer rejects the data channel part of the offer then it doesn't have to do anything as the data channel was not created using the stack. The offerer on the other hand needs to close the data channel that was opened by invoking relevant data channel stack API procedures.

[Editor's note: The information in the preceding paragraph could be moved to Section 6.2.3 (to a new bullet point there).]

It is also worth noting that a data channel stack implementation may not provide any API to create and close data channels; instead the data channels are used on the fly as needed just by communicating via non-DCEP means or by even having some local configuration/assumptions on both the peers.

[Editor's note: The preceding paragraph could be deleted.]

The application then negotiates the data channel properties and sub-protocol properties with the peer's application using a mechanism different from DCEP.

[Editor's note: The preceding paragraph could be deleted.]

[ASSUMPTION] The peer MUST then symmetrically create a data channel with these negotiated data channel properties. This is the only way for the peer's data channel stack to know which properties to apply when transmitting data on this channel. [Editor's note: The previous sentence could be deleted.] The data channel stack MUST allow data channel creation with any non-conflicting stream identifier so that both peers can create the data channel with the same stream identifier. [Editor's note: The information in this sentence could be moved to Section 6.2.3.]

In case the non-DCEP negotiation is correlated with an SDP offer/answer exchange that establishes the SCTP association, the SCTP initialization completion triggers a callback from the data channel stack to an application on both the ends to change the data channel state from Connecting to Open. The details of this interface is specific to the data channel user application. Browser based applications (could include hybrid apps) will use [WebRtcAPI], while native applications use a compatible API, which is yet to be specified. See Section 6.2.3 for details on when the data channel stack can assume the data channel is open, and on when the application can assume the data channel is open.

[Editor's note: The preceding paragraph could be deleted.]

5.2.3. Closing a Data Channel

When the application requests the closing of a non-DCEP negotiated data channel, the data channel stack always performs an SCTP SSN reset for this channel.

[Editor's note: The preceding paragraph could be deleted.]

Depending upon the method used for non-DCEP negotiation and the sub-protocol associated with the data channel, the closing might in addition be signaled to the peer via non-DCEP negotiation.

[Editor's note: The preceding paragraph could be deleted.]

6. SDP Offer/Answer Negotiation

This section defines a method of non-DCEP negotiation by which two clients can negotiate data channel-specific and sub-protocol-specific parameters, using the out-of-band SDP offer/answer exchange. This SDP extension can only be used with the SDP offer/answer model.

6.1. SDP Syntax

Two new SDP attributes are defined to support SDP offer/answer negotiation of data channels. The first attribute provides for negotiation of channel-specific parameters. The second attribute provides for negotiation of sub-protocol-specific parameters.

6.1.1. SDP Attribute for Data Channel Parameter Negotiation

Associated with the SDP "m" line that defines the SCTP association for data channels (defined in Section 4), each SDP offer and answer includes one "a=dcmap:" attribute that defines the data channel parameters for each data channel to be negotiated. Each such attribute line specifies the following parameters for a data channel: SCTP stream identifier, sub-protocol, label, reliability, order of delivery, and priority.

The intention of exchanging these attributes is to create data channels on both the peers with the same set of attributes without actually using the DCEP [I-D.ietf-rtcweb-data-protocol]. It is assumed that the data channel properties (reliable/partially reliable, ordered/unordered) are suitable per the sub-protocol transport requirements.

6.1.1.1. dcmap Attribute

"a=dcmap:" is a media level attribute having following ABNF syntax.

Formal Syntax:

Name: dcmap

Value: dcmap-value

Usage Level: media

Charset Dependent: no

Syntax:

```
dcmap-value     = dcmap-stream-id
                  [ SP dcmap-opt *(";" dcmap-opt) ]
dcmap-opt       = ordering-opt / subprotocol-opt / label-opt
                  / maxretr-opt / maxtime-opt
                  ; Either only maxretr-opt or maxtime-opt
                  ; is present.

dcmap-stream-id = 1*DIGIT
ordering-opt    = "ordered=" ordering-value
ordering-value  = "true" / "false"
subprotocol-opt = "subprotocol=" quoted-string
label-opt       = "label=" quoted-string
maxretr-opt     = "max-retr=" maxretr-value
maxretr-value   = <from-Reliability-Parameter of
                   I-D.ietf-rtcweb-data-protocol>
                  ; number of retransmissions
maxtime-opt     = "max-time=" maxtime-value
maxtime-value   = <from-Reliability-Parameter of
                  I-D.ietf-rtcweb-data-protocol>
                  ; milliseconds

quoted-string   = DQUOTE *(quoted-char / escaped-char) DQUOTE
quoted-char     = SP / quoted-visible
quoted-visible  = %21 / %23-24 / %26-7E ; VCHAR without " or %
escaped-char    = "%" HEXDIG HEXDIG
DQUOTE          = <from-RFC5234>
integer         = <from-RFC5234>
```

Examples:

```
a=dcmap:0
a=dcmap:1 subprotocol="BFCP";max-time=60000
a=dcmap:2 subprotocol="MSRP";ordered=true;label="MSRP"
a=dcmap:3 label="Label 1";ordered=false;max-retr=5
a=dcmap:4 label="foo%09bar";ordered=true;max-time=15000;max-retr=3
```

Note: The last example (a=dcmap:4) shows a 'label' parameter value which contains one non-printable 'escaped-char' character (the tabulator character).

Within an 'a=dcmap' attribute line's 'dcmap-opt' value either only one 'maxretr-opt' parameter or one 'maxtime-opt' parameter MAY be present. Both MUST NOT be present.

6.1.1.2. dcmap-stream-id Parameter

The 'dcmap-stream-id' parameter indicates the SCTP stream identifier within the SCTP association used to form the data channel.

6.1.1.3. label Parameter

The 'label' parameter indicates the name of the channel. It represents a label that can be used to distinguish, in the context of the WebRTC API [WebRtcAPI], an RTCDataChannel object from other RTCDataChannel objects. This parameter maps to the 'Label' parameter defined in [I-D.ietf-rtcweb-data-protocol]. The 'label' parameter is optional. If it is not present, then its value defaults to the empty string.

Note: The empty string MAY also be explicitly used as 'label' value, such that 'label=""' is equivalent to the 'label' parameter not being present at all. [I-D.ietf-rtcweb-data-protocol] allows the DATA_CHANNEL_OPEN message's 'Label' value to be an empty string.

6.1.1.4. subprotocol Parameter

The 'subprotocol' parameter indicates which protocol the client expects to exchange via the channel. 'Subprotocol' is an optional parameter. If the 'subprotocol' parameter is not present, then its value defaults to the empty string.

6.1.1.5. max-retr Parameter

This parameter indicates that the data channel is partially reliable. The 'max-retr' parameter indicates the maximal number a user message will be retransmitted. The max-retr parameter is optional. If the max-retr parameter is not present, then the maximal number of retransmissions is determined as per the generic SCTP retransmission rules as specified in [RFC4960]. This parameter maps to the 'Number of RTX' parameter defined in [I-D.ietf-rtcweb-data-protocol].

6.1.1.6. max-time Parameter

This parameter indicates that the data channel is partially reliable. A user message will no longer be transmitted or retransmitted after a specified life-time given in milliseconds in the 'max-time' parameter. The max-time parameter is optional. If the max-time parameter is not present, then the generic SCTP retransmission timing rules apply as specified in [RFC4960]. This parameter maps to the 'Lifetime in ms' parameter defined in [I-D.ietf-rtcweb-data-protocol].

6.1.1.7. ordered Parameter

The 'ordered' parameter with value "true" indicates that the receiver MUST dispatch DATA chunks in the data channel to the upper layer while preserving the order. The ordered parameter is optional and takes two values: "true" for ordered and "false" for unordered delivery with "true" as the default value. Any other value is ignored and default "ordered=true" is assumed. In the absence of this parameter "ordered=true" is assumed. This parameter maps to the ordered or unordered data channel types as defined in [I-D.ietf-rtcweb-data-protocol].

6.1.2. Sub-Protocol Specific Attributes

In the SDP, each data channel declaration MAY also be followed by other SDP attributes specific to the sub-protocol in use. Each of these attributes is represented by one new attribute line, and it includes the contents of a media-level SDP attribute already defined for use with this (sub)protocol in another IETF specification. Sub-protocol-specific attributes might also be defined for exclusive use with data channel transport, but should use the same syntax described here for other sub-protocol-specific attributes.

Each sub-protocol specific SDP attribute that would normally be used to negotiate the subprotocol using SDP is replaced with an attribute of the form "a=dcsa:stream-id original-attribute", where dcsa stands for "data channel sub-protocol attribute", stream-id is the SCTP stream identifier assigned to this sub-protocol instance, and original-attribute represents the contents of the sub-protocol related attribute to be included.

Formal Syntax:

Name: dcsa

Value: dcsa-value

Usage Level: media

Charset Dependent: no

Syntax:

```
dcsa-value      = stream-id SP attribute
attribute       = <from-RFC4566>
```

Example:

```
a=dcsa:2 accept-types:text/plain
```

Note that the above reference to RFC 4566 defines where the attribute definition can be found; it does not provide any limitation on support of attributes defined in other documents in accordance with this attribute definition. Note however that not all SDP attributes are suitable as "a=dcsa:" parameter. [IANA-SDP-Parameters] contains the lists of IANA registered session and media level or media level only SDP attributes.

Thus in the example above, the original attribute line "a=accept-types:text/plain" is represented by the attribute line "a=dcsa:2 accept-types:text/plain", which specifies that this instance of MSRP being transported on the SCTP association using the data channel with stream id 2 accepts plain text files.

As opposed to the data channel "a=dcmap:" attribute parameters, these parameters are subject to offer/answer negotiation following the procedures defined in the sub-protocol specific documents.

The same syntax applies to any other SDP attribute required for negotiation of this instance of the sub-protocol.

Note: This document does not provide a complete specification of how to negotiate the use of a data channel to transport MSRP. Procedures specific to each sub-protocol such as MSRP will be documented elsewhere. The use of MSRP is only an example of how the generic procedures described herein might apply to a specific sub-protocol.

6.2. Procedures

6.2.1. Managing Stream Identifiers

If an SDP offer/answer exchange (could be the initial or a subsequent one) results in a UDP/DTLS/SCTP or TCP/DTLS/SCTP based media description being accepted, and if this SDP offer/answer exchange results in the establishment of a new SCTP association, then the SDP offerer owns the even SCTP stream ids of this new SCTP association and the answerer owns the odd SCTP stream identifiers. If this "m" line is removed from the signaling session (its port number set to zero), and if usage of this or of a new UDP/DTLS/SCTP or TCP/DTLS/SCTP based "m" line is renegotiated later on, then the even and odd SCTP stream identifier ownership is redetermined as well as described above.

This specification allows simultaneous use of SDP offer/answer and DCEP negotiation. However, an SDP offer/answer exchange MUST NOT be initiated if the associated SCTP stream is already negotiated via DCEP. Stream ids that are not currently used in SDP can be used for DCEP negotiation. Stream id allocation per SDP offer/answer negotiation may not align with DTLS role based allocation. This could cause glare conditions when one side trying to do SDP offer/answer negotiation on a stream id while the other end trying to open a data channel on the same stream id using DCEP negotiation. To avoid these glare conditions this specification recommends that the data channel stack user always selects stream ids per above described SDP offer/answer rule even when DCEP negotiation is used. To avoid glare conditions, it is possible to come up with a different stream id allocation scheme, but such schemes are outside the scope of this specification.

6.2.2. Negotiating Data Channel Parameters

Conveying a reliable data channel is achieved by including neither 'max-retr' nor 'max-time' in corresponding SDP offer's or answer's a=dcmap attribute line. Conveying a partially reliable data channel is achieved by including only one of 'max-retr' or 'max-time'. By definition max-retr and max-time are mutually exclusive, so at most one of them MAY be present in a=dcmap. If an SDP offer contains both of these parameters then the receiver of such an SDP offer MUST reject the SDP offer. If an SDP answer contains both of these parameters then the offerer MAY treat it as an error and MAY assume the associated SDP offer/answer failed and MAY take appropriate recovery actions. These recovery options are outside the scope of this specification.

The SDP answer SHALL echo the same subprotocol, max-retr, max-time, ordered parameters, if those were present in the offer, and MAY include a label parameter. They MAY appear in any order, which could be different from the SDP offer, in the SDP answer.

When sending a subsequent offer or an answer, and for as long as the data channel is still open, the sender MUST replicate the same information.

Data channel types defined in [I-D.ietf-rtcweb-data-protocol] are mapped to SDP in the following manner, where "ordered=true" is the default and may be omitted:

DATA_CHANNEL_RELIABLE
ordered=true

DATA_CHANNEL_RELIABLE_UNORDERED
ordered=false

DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT
ordered=true;max-retr=<number of retransmissions>

DATA_CHANNEL_PARTIAL_RELIABLE_REXMIT_UNORDERED
ordered=false;max-retr=<number of retransmissions>

DATA_CHANNEL_PARTIAL_RELIABLE_TIMED
ordered=true;max-time=<lifetime in milliseconds>

DATA_CHANNEL_PARTIAL_RELIABLE_TIMED_UNORDERED
ordered=false;max-time=<lifetime in milliseconds>

6.2.3. Opening a Data Channel

The procedure for opening a data channel using SDP offer/answer negotiation starts with the agent preparing to send an SDP offer. If a peer receives an SDP offer before starting to send a new SDP offer with data channels that are to be SDP offer/answer negotiated, or loses an SDP offer glare resolution procedure in this case, it MUST wait until the ongoing SDP offer/answer completes before resuming the SDP offer/answer negotiation procedure.

The agent that intends to send an SDP offer to create data channels through SDP offer/answer negotiation performs the following:

- Creates data channels using stream identifiers from the owned set (see Section 6.2.1).

- Generates a new SDP offer.

- Determines the list of stream identifiers assigned to data channels opened through SDP offer/answer negotiation.

- Completes the SDP offer with the dcmap and dcsa attributes needed, if any, for each SDP offer/answer negotiated data channel, as described in Section 6.1 and in Section 6.2.2.

- Sends the SDP offer.

The peer receiving such an SDP offer performs the following:

- Parses and applies the SDP offer. Note that the typical parser normally ignores unknown SDP attributes, which includes data channel related attributes.

- Analyzes the channel parameters and sub-protocol attributes to determine whether to accept each offered data channel.

- For accepted data channels, it creates peer instances for the data channels with the agent using the channel parameters described in the SDP offer. Note that the agent is asked to create data channels with SCTP stream identifiers contained in the SDP offer if the SDP offer is accepted.

- Generates an SDP answer.

- Completes the SDP answer with the dcmap and optional dcsa attributes needed for each SDP offer/answer negotiated data channel, as described in Section 6.1 and in Section 6.2.2.

- Sends the SDP answer.

The agent receiving such an SDP answer performs the following:

- Closes any created data channels for which the expected dcmap and dcsa attributes are not present in the SDP answer.

- Applies the SDP answer.

Each agent application MUST wait to send data until it has confirmation that the data channel at the peer is instantiated. For WebRTC, this is when both data channel stacks have channel parameters instantiated. This occurs:

- At both peers when a data channel is created without an established SCTP association, as soon as the SCTP association is successfully established.

- At the agent receiving an SDP offer for which there is an established SCTP association, as soon as it creates an SDP offer/answer negotiated data channel based on information signaled in the SDP offer.

- At the agent sending an SDP offer to create a new SDP offer/answer negotiated data channel for which there is an established SCTP association, when it receives the SDP answer confirming acceptance of the data channel or when it begins to receive data on the data channel from the peer, whichever occurs first.

6.2.4. Closing a Data Channel

When the application requests the closing of a data channel that was negotiated via SDP offer/answer, the data channel stack always performs an SCTP SSN reset for this channel.

It is specific to the sub-protocol whether this closing MUST in addition be signaled to the peer via a new SDP offer/answer exchange.

The intention to close a data channel can be signaled by sending a new SDP offer which excludes the "a=dcmap:" and "a=dcsa:" attribute lines for the data channel. The offerer SHOULD NOT change the port value for the "m" line (e.g. to zero) when closing a data channel (unless all data channels are being closed and the SCTP association is no longer needed), since this would close the SCTP association and impact all of the data channels. If the answerer accepts the SDP offer then the answerer MUST close those data channels whose "a=dcmap:" and "a=dcsa:" attribute lines were excluded from the received SDP offer, unless those data channels were already closed, and the answerer MUST also exclude the corresponding attribute lines in the answer. In addition to that, the SDP answerer MAY exclude other data channels which were closed but not yet communicated to the peer. So, the offerer MUST inspect the answer to see if it has to close other data channels which are now not included in the answer.

If a new SDP offer/answer is used to close data channels then the data channel(s) SHOULD only be closed by the answerer/offerer after a successful SDP answer is sent/received.

> This delayed closure is RECOMMENDED in order to handle cases where a successful SDP answer is not received, in which case the state of the session SHOULD be kept per the last successful SDP offer/answer.

If a client receives a data channel close indication (due to inband SCTP SSN reset or some other reason) without associated SDP offer

then the client SHOULD generate an SDP offer which excludes this closed data channel.

The application MUST also close any data channel that was negotiated via SDP offer/answer, for which the stream identifiers are not listed in an incoming SDP offer.

A closed data channel using local close (SCTP SSN reset), without an additional SDP offer/answer to close it, may be reused for a new data channel. This can only be done via new SDP offer/answer, describing the new sub-protocol and its attributes, only after the corresponding data channel close acknowledgement is received from the peer (i.e. SCTP SSN reset of both incoming and outgoing streams is completed). This restriction is to avoid the race conditions between arrival of "SDP offer which reuses stream" with "SCTP SSN reset which closes outgoing stream" at the peer.

6.2.5. Various SDP Offer/Answer Scenarios and Considerations

SDP offer has no a=dcmap attributes

* Initial SDP offer: No data channel is negotiated yet. The DTLS connection and SCTP association is negotiated and, if agreed, established as per [I-D.ietf-mmusic-sctp-sdp].

* Subsequent SDP offer: All the SDP offer/answer negotiated data channels are expected be closed now. The DTLS/SCTP association remains open for SDP offer/answer or DCEP negotiation of data channels.

SDP answer has no a=dcmap attributes

* Initial SDP answer: Either the peer does not support dcmap attributes or it rejected all the data channels. In either case the offerer closes all the SDP offer/answer negotiated data channels that were open at the time of initial offer. The DTLS connection and SCTP association will still be setup.

* Subsequent SDP answer: All the SDP offer/answer negotiated data channels are expected be closed now. The DTLS/SCTP association remains open for future SDP offer/answer or DCEP negotiation of data channels.

SDP offer has no a=dcsa attributes for a data channel.

* This is allowed and indicates there are no sub-protocol parameters to convey.

SDP answer has no a=dcsa attributes for a data channel.

* This is allowed and indicates there are no sub-protocol parameters to convey in the SDP answer. The number of dcsa attributes in the SDP answer does not have to match the number of dcsa attributes in the SDP offer.

7. Examples

```
SDP offer:
  m=application 10001 UDP/DTLS/SCTP webrtc-datachannel
  c=IN IP4 10.10.10.1
  a=max-message-size:100000
  a=sctp-port 5000
  a=setup:actpass
  a=connection:new
  a=fingerprint:SHA-1 \
      4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
  a=dcmap:0 subprotocol="BFCP";label="BFCP"

SDP answer:
  m=application 10002 UDP/DTLS/SCTP webrtc-datachannel
  c=IN IP4 10.10.10.2
  a=max-message-size:100000
  a=sctp-port 5002
  a=setup:passive
  a=connection:new
  a=fingerprint:SHA-1 \
      5B:AD:67:B1:3E:82:AC:3B:90:02:B1:DF:12:5D:CA:6B:3F:E5:54:FA
```

Figure 1: Example 1

In the above example the SDP answerer rejected the data channel with stream id 0 either for explicit reasons or because it does not understand the a=dcmap attribute. As a result the offerer will close the data channel created with the SDP offer/answer negotiation option. The SCTP association will still be setup over DTLS. At this point the offerer or the answerer may use DCEP negotiation to open data channels.

```
SDP offer:
  m=application 10001 UDP/DTLS/SCTP webrtc-datachannel
  c=IN IP4 10.10.10.1
  a=max-message-size:100000
  a=sctp-port 5000
  a=setup:actpass
  a=connection:new
  a=fingerprint:SHA-1 \
      4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
  a=dcmap:0 subprotocol="BFCP";label="BFCP"
  a=dcmap:2 subprotocol="MSRP";label="MSRP"
  a=dcsa:2 accept-types:message/cpim text/plain text/
  a=dcsa:2 path:msrp://alice.example.com:10001/2s93i93idj;dc

SDP answer:
  m=application 10002 UDP/DTLS/SCTP webrtc-datachannel
  c=IN IP4 10.10.10.2
  a=max-message-size:100000
  a=sctp-port 5002
  a=setup:passive
  a=connection:new
  a=fingerprint:SHA-1 \
      5B:AD:67:B1:3E:82:AC:3B:90:02:B1:DF:12:5D:CA:6B:3F:E5:54:FA
  a=dcmap:2 subprotocol="MSRP";label="MSRP"
  a=dcsa:2 accept-types:message/cpim text/plain
  a=dcsa:2 path:msrp://bob.example.com:10002/si438dsaodes;dc
```

Figure 2: Example 2

In the above example the SDP offer contains data channels for BFCP and MSRP sub-protocols. The SDP answer rejected BFCP and accepted MSRP. So, the offerer should close the data channel for BFCP and both offerer and answerer may start using the MSRP data channel (after SCTP/DTLS association is setup). The data channel with stream id 0 is free and can be used for future DCEP or SDP offer/answer negotiation.

Continuing on the earlier example in Figure 1.

```
Subsequent SDP offer:
  m=application 10001 UDP/DTLS/SCTP webrtc-datachannel
  c=IN IP4 10.10.10.1
  a=max-message-size:100000
  a=sctp-port 5000
  a=setup:actpass
  a=connection:existing
  a=fingerprint:SHA-1 \
      4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
  a=dcmap:4 subprotocol="MSRP";label="MSRP"
  a=dcsa:4 accept-types:message/cpim text/plain
  a=dcsa:4 path:msrp://alice.example.com:10001/2s93i93idj;dc

Subsequent SDP answer:
  m=application 10002 UDP/DTLS/SCTP webrtc-datachannel
  c=IN IP4 10.10.10.2
  a=max-message-size:100000
  a=sctp-port 5002
  a=setup:passive
  a=connection:existing
  a=fingerprint:SHA-1 \
      5B:AD:67:B1:3E:82:AC:3B:90:02:B1:DF:12:5D:CA:6B:3F:E5:54:FA
  a=dcmap:4 subprotocol="MSRP";label="MSRP"
  a=dcsa:4 accept-types:message/cpim text/plain
  a=dcsa:4 path:msrp://bob.example.com:10002/si438dsaodes;dc
```

Figure 3: Example 3

The above example is a continuation of the example in Figure 1. The SDP offer now removes the MSRP data channel with stream id 2, but opens a new MSRP data channel with stream id 4. The answerer accepts the entire offer. As a result the offerer closes the earlier negotiated MSRP related data channel and both offerer and answerer may start using new the MSRP related data channel.

8. Security Considerations

No security considerations are envisaged beyond those already documented in [RFC4566].

9. IANA Considerations

9.1. Subprotocol Identifiers

Registration of new subprotocol identifiers is performed using the existing IANA table "WebSocket Subprotocol Name Registry".

The following text should be added following the title of the table.

"This table also includes subprotocol identifiers specified for usage within a WebRTC data channel."

The following reference should be added to under the heading reference: "RFC XXXX".

This document assigns no new values to this table.

NOTE to RFC Editor: Please replace "XXXX" with the number of this RFC.

9.2. New SDP Attributes

9.2.1. dcmap

[Editor's note: This section still needs to be completed.]

9.2.2. dcsa

[Editor's note: This section still needs to be completed.]

10. Acknowledgments

The authors wish to acknowledge the borrowing of ideas from other internet drafts by Salvatore Loreto, Gonzalo Camarillo, Peter Dunkley and Gavin Llewellyn, and to thank Roni Even, Christian Groves, Christer Holmberg, Paul Kyzivat, Jonathan Lennox, and Uwe Rauschenbach for their invaluable comments.

11. CHANGE LOG

11.1. Changes against 'draft-ietf-mmusic-data-channel-sdpneg-02'

- Move of reference [I-D.ietf-rtcweb-jsep] from the list of normative references to the list of informative references.

- Addition of [IANA-SDP-Parameters] to the list of informative references and addition of following two sentences to the first paragraph after the ABNF definition: "Note however that not all SDP attributes are suitable as "a=dcsa:" parameter. [IANA-SDP-Parameters] contains the lists of IANA registered session and media level or media level only SDP attributes."

- In the introduction replacement of last sentence "This document defines SDP-based out-of-band negotiation procedures to establish data channels for transport of well-defined sub-protocols" with "This document defines SDP offer/answer negotiation procedures to

establish data channels for transport of well-defined sub-protocols, to enable out-of-band negotiation".

- Throughout the document replacement of "external negotiation" with "SDP offer/answer negotiation" and removal of term "external negotiation" from the terminology list in Section 3.

- Throughout the document replacement of "internal negotiation" with "DCEP" and removal of terms "internal negotiation" and "in-band negotiation" from the terminology list in Section 3.

- Addition of "SCTP Stream Sequence Number (SSN)" to the list of terms.

- In Section 6.2.1 replacement of sentence "However, a single stream is managed using one method at a time." with "However, an SDP offer/answer exchange MUST NOT be initiated if the associated SCTP stream is already negotiated via DCEP".

- In Section 6.2.2 replacement of sentence "By definition max-retr and max-time are mutually exclusive, so only one of them can be present in a=dcmap" with "By definition max-retr and max-time are mutually exclusive, so at most one of them MAY be present in a=dcmap".

- Move of reference [WebRtcAPI] from list of normative references to list of informative references.

- Removal of almost all text parts, which discussed JavaScript or other API specific aspects. Such API specific aspects were mainly discussed in sub-sections of Section 5 and Section 6.

11.2. Changes against 'draft-ietf-mmusic-data-channel-sdpneg-01'

- New Section 4 regarding applicability to SDP offer/answer only.

- Addition of new Section 9.1 "Subprotocol identifiers" as subsection of the "IANA Considerations" related Section 9. Also removal of the temporary note "To be completed. As [I-D.ietf-rtcweb-data-protocol] this document should refer to IANA's WebSocket Subprotocol Name Registry defined in [RFC6455]."

- In Section 6.2.2:

  * In the first paragraph replacement of the sentence "If an SDP offer contains both of these parameters then such an SDP offer will be rejected." with "If an SDP offer contains both of these

parameters then the receiver of such an SDP offer MUST reject the SDP offer."

* In the second paragraph capitalization of "shall" and "may" such that both sentences now read: "The SDP answer SHALL echo the same subprotocol, max-retr, max-time, ordered parameters, if those were present in the offer, and MAY include a label parameter. They MAY appear in any order, which could be different from the SDP offer, in the SDP answer."

* In the third paragraph replacement of the sentence "The same information MUST be replicated without changes in any subsequent offer or answer, as long as the data channel is still opened at the time of offer or answer generation." with "When sending a subsequent offer or an answer, and for as long as the data channel is still open, the sender MUST replicate the same information.".

o In Section 6.2.2 the mapping of data channel types defined in [I-D.ietf-rtcweb-data-protocol] to the SDP "a=dcmap" attribute parameters were illustrated using example "a=dcmap" attribute lines. Replacement of these example "a=dcmap" attribute lines with just the "a=dcmap" attribute parameters being relevant for the channel type.

o In Section 6.2.5 the description of bullet point "SDP offer has no a=dcmap attributes - Initial SDP offer:" was "Initial SDP offer: No data channel negotiated yet." Replacement of this description with "Initial SDP offer: No data channel is negotiated yet. The DTLS connection and SCTP association is negotiated and, if agreed, established as per [I-D.ietf-mmusic-sctp-sdp]."

o In Section 6.2.5 in both bullet points related to "Subsequent SDP offer" and "Subsequent SDP answer" replacement of "All the externally negotiated data channels must be closed now." with "All the externally negotiated data channels are expected be closed now.".

o In Section 5.2.2's sixth paragraph beginning with "[ASSUMPTION]" replacement of the two occurrences of "must" with "MUST".

o In Section 6.1.1.1 in the definition of the ABNF rule "dcmap-opt" there was a comment saying that "Either only maxretr-opt or maxtime-opt is present. Both MUST not be present." Removal of the second normative sentence and instead addition of following new paragraph to the end of this section: "Within an 'a=dcmap' attribute line's 'dcmap-opt' value either only one 'maxretr-opt'

parameter or one 'maxtime-opt' parameter is present. Both MUST NOT be present."

- In Section 6.1.1.7 replacement of the first sentence "The 'ordered' parameter with value "true" indicates that DATA chunks in the channel MUST be dispatched to the upper layer by the receiver while preserving the order." with "The 'ordered' parameter with value "true" indicates that the receiver MUST dispatch DATA chunks in the data channel to the upper layer while preserving the order.".

- In Section 6.2.3's first paragraph replacement of the one occurrence of "must" with "..., it MUST wait until ...".

- In Section 6.2.4:

  * In the second paragraph replacement of "must" with "... whether this closing MUST in addition ..."

  * In the third paragraph replacement of the sentence "The port value for the "m" line SHOULD not be changed (e.g., to zero) when closing a data channel ..." with "The offerer SHOULD NOT change the port value for the "m" line (e.g., to zero) when closing a data channel ...".

  * In the last but two paragraph replacement of the sentence "... then an SDP offer which excludes this closed data channel SHOULD be generated." with "... then the client SHOULD generate an SDP offer which excludes this closed data channel.".

  * In the last but one paragraph replacement of "must" with "The application MUST also close...".

- In Section 6.1.2 addition of following note after the formal definition of the 'a=dcsa' attribute: "Note that the above reference to RFC 4566 defines were the attribute definition can be found; it does not provide any limitation on support of attributes defined in other documents in accordance with this attribute definition."

11.3. Changes against 'draft-ietf-mmusic-data-channel-sdpneg-00'

- In Section 3 "WebRTC data channel" was defined as "A bidirectional channel consisting of paired SCTP outbound and inbound streams." Replacement of this definition with "Data channel: A WebRTC data channel as specified in [I-D.ietf-rtcweb-data-channel]", and consistent usage of "data channel" in the remainder of the document including the document's headline."

- In Section 5 removal of following note: 'OPEN ISSUE: The syntax in [I-D.ietf-mmusic-sctp-sdp] may change as that document progresses. In particular we expect "webrtc-datachannel" to become a more general term.'

- Consistent usage of '"m" line' in whole document as per [RFC4566].

- In Section 6.1.1 removal of the example dcmap attribute line 'a=dcmap:2 subprotocol="BFCP";label="channel 2' as there are already four examples right after the ABNF rules in Section 6.1.1.1. Corresponding removal of following related note: "Note: This document does not provide a complete specification of how to negotiate the use of a WebRTC data channel to transport BFCP. Procedures specific to each sub-protocol such as BFCP will be documented elsewhere. The use of BFCP is only an example of how the generic procedures described herein might apply to a specific sub-protocol."

- In Section 6.1.1 removal of following note: "Note: This attribute is derived from attribute "webrtc-DataChannel", which was defined in old version 03 of the following draft, but which was removed along with any support for SDP external negotiation in subsequent versions: [I-D.ietf-mmusic-sctp-sdp]."

- Insertion of following new sentence to the beginning of Section 6.1.1.1: "dcmap is a media level attribute having following ABNF syntax:"

- Insertion of new Section 6.1.1.2 containing the dcmap-stream-id specifying sentence, which previously was placed right before the formal ABNF rules. Removal of the sentence 'Stream is a mandatory parameter and is noted directly after the "a=dcmap:" attribute's colon' as this information is part of the ABNF specification.

- In Section 6.1.1.1 modification of the 'ordering-value' values from "0" or "1" to "true" or "false". Corresponding text modifications in Section 6.1.1.7.

- In Section 6.1.1.1 the ABNF definition of "quoted-string" referred to rule name "escaped-char", which was not defined. Instead a rule with name "escaped" was defined. Renamed that rule's name to "escaped-char".

- Insertion of a dedicated note right after the "a=dcmap:4" attribute example in Section 6.1.1.1 regarding the non-printable "escaped-char" character within the "label" value.

- In Section 6.1.2's second paragraph replacement of "sctp stream identifier" with "SCTP stream identifier".

- In first paragraph of Section 6.2.1 replacement of first two sentences 'For the SDP-based external negotiation described in this document, the initial offerer based "SCTP over DTLS" owns by convention the even stream identifiers whereas the initial answerer owns the odd stream identifiers. This ownership is invariant for the whole lifetime of the signaling session, e.g. it does not change if the initial answerer sends a new offer to the initial offerer.' with 'If an SDP offer/answer exchange (could be the initial or a subsequent one) results in a UDP/DTLS/SCTP or TCP/DTLS/SCTP based media description being accepted, and if this SDP offer/answer exchange results in the establishment of a new SCTP association, then the SDP offerer owns the even SCTP stream ids of this new SCTP association and the answerer owns the odd SCTP stream identifiers. If this "m" line is removed from the signaling session (its port number set to zero), and if usage of this or of a new UDP/DTLS/SCTP or TCP/DTLS/SCTP based "m" line is renegotiated later on, then the even and odd SCTP stream identifier ownership is redetermined as well as described above.'

- In Section 6.2.3 the first action of an SDP answerer, when receiving an SDP offer, was described as "Applies the SDP offer. Note that the browser ignores data channel specific attributes in the SDP." Replacement of these two sentences with "Parses and applies the SDP offer. Note that the typical parser normally ignores unknown SDP attributes, which includes data channel related attributes."

- In Section 6.2.3 the second sentence of the third SDP answerer action was "Note that the browser is asked to create data channels with stream identifiers not "owned" by the agent.". Replacement of this sentence with "Note that the agent is asked to create data channels with SCTP stream identifiers contained in the SDP offer if the SDP offer is accepted."

- In Section 6.2.4 the third paragraph began with "A data channel can be closed by sending a new SDP offer which excludes the dcmap and dcsa attribute lines for the data channel. The port value for the m line should not be changed (e.g., to zero) when closing a data channel (unless all data channels are being closed and the SCTP association is no longer needed), since this would close the SCTP association and impact all of the data channels. If the answerer accepts the SDP offer then it MUST also exclude the corresponding attribute lines in the answer. ..." Replacement of this part with "The intention to close a data channel can be signaled by sending a new SDP offer which excludes the "a=dcmap:"

and "a=dcsa:" attribute lines for the data channel. The port value for the "m" line SHOULD not be changed (e.g., to zero) when closing a data channel (unless all data channels are being closed and the SCTP association is no longer needed), since this would close the SCTP association and impact all of the data channels. If the answerer accepts the SDP offer then it MUST close those data channels whose "a=dcmap:" and "a=dcsa:" attribute lines were excluded from the received SDP offer, unless those data channels were already closed, and it MUST also exclude the corresponding attribute lines in the answer."

- In Section 6.2.4 the hanging text after the third paragraph was "This delayed close is to handle cases where a successful SDP answer is not received, in which case the state of session should be kept per the last successful SDP offer/answer." Replacement of this sentence with "This delayed closure is RECOMMENDED in order to handle cases where a successful SDP answer is not received, in which case the state of the session SHOULD be kept per the last successful SDP offer/answer."

- Although dedicated to "a=dcmap" and "a=dcsa" SDP syntax aspects Section 6.1.1 contained already procedural descriptions related to data channel reliability negotiation. Creation of new Section 6.2.2 and moval of reliability negotiation related text to this new section.

11.4. Changes against 'draft-ejzak-mmusic-data-channel-sdpneg-02'

- Removal of note "[ACTION ITEM]" from section "subprotocol parameter". As [I-D.ietf-rtcweb-data-protocol] this document should refer to IANA's WebSocket Subprotocol Name Registry defined in [RFC6455].

- In whole document, replacement of "unreliable" with "partially reliable", which is used in [I-D.ietf-rtcweb-data-channel] and in [I-D.ietf-rtcweb-data-protocol] in most places.

- Clarification of the semantic if the "max-retr" parameter is not present in an a=dcmap attribute line. In section "max-retr parameter" the sentence "The max-retr parameter is optional with default value unbounded" was replaced with "The max-retr parameter is optional. If the max-retr parameter is not present, then the maximal number of retransmissions is determined as per the generic SCTP retransmission rules as specified in [RFC4960]".

- Clarification of the semantic if the "max-time" parameter is not present in an a=dcmap attribute line. In section "max-time parameter" the sentence "The max-time parameter is optional with

default value unbounded" was replaced with "The max-time parameter is optional. If the max-time parameter is not present, then the generic SCTP retransmission timing rules apply as specified in [RFC4960]".

- In section "label parameter" the sentence "Label is a mandatory parameter." was removed and following new sentences (including the note) were added: "The 'label' parameter is optional. If it is not present, then its value defaults to the empty string. Note: The empty string may also be explicitly used as 'label' value, such that 'label=""' is equivalent to the 'label' parameter not being present at all. [I-D.ietf-rtcweb-data-protocol] allows the DATA_CHANNEL_OPEN message's 'Label' value to be an empty string."

- In section "subprotocol parameter" the sentence "Subprotocol is a mandatory parameter." was replaced with "'Subprotocol' is an optional parameter. If the 'subprotocol' parameter is not present, then its value defaults to the empty string."

- In the "Examples" section, in the first two SDP offer examples in the a=dcmap attribute lines 'label="BGCP"' was replaced with 'label="BFCP"'.

- In all examples, the "m" line proto value "DTLS/SCTP" was replaced with "UDP/DTLS/SCTP" and the "a=fmtp" attribute lines were replaced with "a=max-message-size" attribute lines, as per draft-ietf-mmusic-sctp-sdp-12.

11.5. Changes against '-01'

- Formal syntax for dcmap and dcsa attribute lines.

- Making subprotocol as an optional parameter in dcmap.

- Specifying disallowed parameter combinations for max-time and max-retr.

- Clarifications on WebRTC data channel close procedures.

11.6. Changes against '-00'

- Revisions to identify difference between internal and external negotiation and their usage.

- Introduction of more generic terminology, e.g. "application" instead of "browser".

- Clarification of how "max-retr and max-time affect the usage of unreliable and reliable WebRTC data channels.

- Updates of examples to take into account the SDP syntax changes introduced with draft-ietf-mmusic-sctp-sdp-07.

- Removal of the SCTP port number from the a=dcmap and a=dcsa attributes as this is now contained in the a=sctp-port attribute, and as draft-ietf-mmusic-sctp-sdp-07 supports only one SCTP association on top of the DTLS connection.

12. References

12.1. 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>.

[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>.

[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>.

[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>.

[I-D.ietf-rtcweb-data-channel] Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channels", draft-ietf-rtcweb-data-channel-13 (work in progress), January 2015.

[I-D.ietf-mmusic-sctp-sdp] Holmberg, C., Loreto, S., and G. Camarillo, "Stream Control Transmission Protocol (SCTP)-Based Media Transport in the Session Description Protocol (SDP)", draft-ietf-mmusic-sctp-sdp-14 (work in progress), March 2015.

12.2. Informative References

[I-D.ietf-rtcweb-data-protocol]
Jesup, R., Loreto, S., and M. Tuexen, "WebRTC Data Channel Establishment Protocol", draft-ietf-rtcweb-data-protocol-09 (work in progress), January 2015.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <http://www.rfc-editor.org/info/rfc4960>.

[RFC4975] Campbell, B., Ed., Mahy, R., Ed., and C. Jennings, Ed., "The Message Session Relay Protocol (MSRP)", RFC 4975, DOI 10.17487/RFC4975, September 2007, <http://www.rfc-editor.org/info/rfc4975>.

[RFC4976] Jennings, C., Mahy, R., and A. Roach, "Relay Extensions for the Message Sessions Relay Protocol (MSRP)", RFC 4976, DOI 10.17487/RFC4976, September 2007, <http://www.rfc-editor.org/info/rfc4976>.

[RFC5547] Garcia-Martin, M., Isomaki, M., Camarillo, G., Loreto, S., and P. Kyzivat, "A Session Description Protocol (SDP) Offer/Answer Mechanism to Enable File Transfer", RFC 5547, DOI 10.17487/RFC5547, May 2009, <http://www.rfc-editor.org/info/rfc5547>.

[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, <http://www.rfc-editor.org/info/rfc6135>.

[RFC6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, DOI 10.17487/RFC6455, December 2011, <http://www.rfc-editor.org/info/rfc6455>.

[RFC6714] Holmberg, C., Blau, S., and E. Burger, "Connection Establishment for Media Anchoring (CEMA) for the Message Session Relay Protocol (MSRP)", RFC 6714, DOI 10.17487/RFC6714, August 2012, <http://www.rfc-editor.org/info/rfc6714>.

[I-D.ietf-rtcweb-jsep]
Uberti, J., Jennings, C., and E. Rescorla, "Javascript Session Establishment Protocol", draft-ietf-rtcweb-jsep-11 (work in progress), July 2015.

[IANA-SDP-Parameters]
"Session Description Protocol (SDP) Parameters", Internet Assigned Numbers Authority Protocol Assignments Session Description Protocol (SDP) Parameters, <http://www.iana.org/assignments/sdp-parameters/sdp-parameters.xhtml>.

[WebRtcAPI]
Bergkvist, A., Burnett, D., Jennings, C., and A. Narayanan, "WebRTC 1.0: Real-time Communication Between Browsers", World Wide Web Consortium WD-webrtc-20150210, February 2015, <http://www.w3.org/TR/2015/WD-webrtc-20150210/>.

Authors' Addresses

Keith Drage (editor)
Alcatel-Lucent
Quadrant, Stonehill Green, Westlea
Swindon
UK

Email: keith.drage@alcatel-lucent.com

Maridi R. Makaraju (Raju)
Alcatel-Lucent
2000 Lucent Lane
Naperville, Illinois
US

Email: Raju.Makaraju@alcatel-lucent.com

Juergen Stoetzer-Bradler
Alcatel-Lucent
Lorenzstrasse 10
D-70435 Stuttgart
Germany

Email: Juergen.Stoetzer-Bradler@alcatel-lucent.com

Richard Ejzak
Unaffiliated

Email: richard.ejzak@gmail.com

Jerome Marcon
Unaffiliated