T.140 Real-time Text Conversation over WebRTC Data Channels
draft-ietf-mmusic-t140-usage-data-channel-09

Abstract

   This document specifies how a WebRTC data channel can be used as a
   transport mechanism for Real-time text using the ITU-T Protocol for
   multimedia application text conversation (Recommendation ITU-T
   T.140), and how the SDP offer/answer mechanism can be used to
   negotiate such data channel, referred to as T.140 data channel. The
   document updates RFC 8373.

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 June 1, 2020.

the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction

   The ITU-T Protocol for multimedia application text conversation
   (Recommendation ITU-T T.140) [T140] defines a protocol for text
   conversation, also known as real-time text. The transport used for
   IP networks is the "RTP Payload for Text Conversation" [RFC4103]
   mechanism, based on the Real-time Transport Protocol (RTP) [RFC3550].

   This document specifies how a WebRTC data channel
   [I-D.ietf-rtcweb-data-channel] can be used as a transport mechanism
   for T.140, and how the SDP offer/answer mechanism
   [I-D.ietf-mmusic-data-channel-sdpneg] can be used to negotiate such
   data channel.

In this document, a T.140 data channel refers to a WebRTC data
channel for which the instantiated sub-protocol is "t140", and where
the channel is negotiated using the SDP-based external negotiation
method [I-D.ietf-mmusic-data-channel-sdpneg].

NOTE: The decision to transport real-time text using a WebRTC data
channel, instead of using RTP based transport [RFC4103], is motivated
by use-case "U-C 5: Real-time text chat during an audio and/or video
call with an individual or with multiple people in a conference", see
Section 3.2 of [I-D.ietf-rtcweb-data-channel].

The brief notation "T.140" is used as a name for the text
conversation protocol according to [T140].

Section 3 defines the generic data channel properties for a T.140
data channel, and Section 4 defines how they are conveyed in an SDP
dcmap attribute. While this document defines how to establish a
T.140 data channel using the SDP-based external negotiation method
[I-D.ietf-mmusic-data-channel-sdpneg], the generic T.140 and gateway
considerations defined in Section 3, Section 5 and Section 6 of this
document can also be applied when a T.140 data channel is established
using another mechanism (e.g., the mechanism defined in
[I-D.ietf-rtcweb-data-protocol]). Section 5 of
[I-D.ietf-mmusic-data-channel-sdpneg] defines the mapping between the
SDP dcmap attribute parameters and the protocol parameters used in
[I-D.ietf-rtcweb-data-protocol].

This document is based on an earlier Internet draft edited by Keith
Drage, Juergen Stoetzer-Bradler and Albrecht Schwarz.

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. WebRTC Data Channel Considerations

The following WebRTC data channel property values
[I-D.ietf-rtcweb-data-channel] apply to a T.140 data channel:

```
+--------------------------+------------------------------+
| Subprotocol Identifier   | t140                         |
| Transmission reliability | reliable                     |
| Transmission order       | in-order                     |
| Label                    | See Section 4.1 and Section 6 |
+--------------------------+------------------------------+
```

NOTE: T.140 requires the transport channel to provide transmission of
real-time text without duplication and in original order. Therefore,
T.140 does not specify reliable and ordered transmission of T.140
data on the application layer. Instead, when RTP based transport is
used, the RTP sequence number is used to detect packet loss and out-
of-order packets, and a redundancy mechanism is used to achieve
reliable delivery of T.140 data. By using the WebRTC data channel
reliable and in-order transmission features
[I-D.ietf-rtcweb-data-channel] for the T.140 data channel, there is
no need for a redundancy mechanism or a mechanism to detect data loss
and out-of-order delivery at the application level. The latency
characteristics of the T.140 data channel is also regarded to be
sufficient to meet the application requirements of T.140.

4. SDP Considerations

The generic SDP considerations, including the SDP Offer/Answer
procedures, for negotiating a WebRTC data channel are defined in
[I-D.ietf-mmusic-data-channel-sdpneg]. This section defines the SDP
considerations that are specific to a T.140 data channel.

4.1. Use of dcmap Attribute

An offerer and answerer MUST, in each offer and answer, include an
SDP 'dcmap' attribute [I-D.ietf-mmusic-data-channel-sdpneg] in the
SDP media description (m= section) [I-D.ietf-mmusic-rfc4566bis]
describing the SCTP association [RFC4960] used to realize the T.140
data channel.

The offerer and answerer MUST include the subprotocol attribute
parameter, with a "t140" parameter value, in the 'dcmap' attribute
value.

The offerer and answerer MAY include the priority attribute parameter
and the label attribute parameter in the 'dcmap' attribute value. If
the offerer includes a label attribute parameter, the answerer MUST
NOT change the value in the answer.

NOTE: As specified in [I-D.ietf-rtcweb-data-channel], when a data
channel is negotiated using the mechanism defined in
[I-D.ietf-rtcweb-data-protocol], the label attribute parameter value

has to be the same in both directions. That rule also applies to
data channels negotiated using the mechanism defined in this
document.

The offerer and answerer MUST NOT include the max-retr and max-time
attribute parameters in the 'dcmap' attribute.

If the ordered attribute parameter is included in the 'dcmap'
attribute, it MUST be assigned the value 'true'.

Below is an example of the 'dcmap' attribute for a T.140 data channel
with stream id=3 and without any label:

a=dcmap:3 subprotocol="t140"

4.2. Use of dcsa Attribute

An offerer and answerer MAY, in each offer and answer, include one or
more SDP 'dcsa' attributes [I-D.ietf-mmusic-data-channel-sdpneg] in
the m= section describing the SCTP association used to realize the
T.140 data channel.

If an offerer or answerer receives a 'dcsa' attribute that contains
an SDP attribute which usage has not been defined for a T.140 data
channel, the offerer or answerer should ignore the 'dcsa' attribute,
following the rules in Section 6.7 of
[I-D.ietf-mmusic-data-channel-sdpneg].

4.2.1. Maximum Character Transmission Rate

A 'dcsa' attribute can contain the SDP 'fmtp' attribute used to
indicate a maximum character transmission rate [RFC4103]. The 'cps'
attribute parameter is used to indicate the maximum character
transmission rate that the endpoint that includes the attribute is
able to receive, and the value is used as a mean value in characters
per second over any 10-second interval.

If the 'fmtp' attribute is included, the 'format' attribute parameter
MUST be set to "-".

If the 'fmtp' attribute is not included, the default value of 30
applies [RFC4103].

The offerer and answerer MAY modify the 'cps' attribute parameter
value in subsequent offers and answers.

This document does not define any other usage of the 'fmtp' attribute
for a T.140 channel. If an offerer or answerer receives a 'dcsa'

attribute that contains an 'fmtp' attribute that is not according to
the procedure above, the offerer or answerer MUST ignore the 'dcsa'
attribute.

NOTE: The 'cps' attribute parameter is especially useful when a T.140
data channel endpoint is acting as a gateway (Section 6) and is
interworking with a T.140 transport mechanism that have restrictions
on how many characters can be sent per second.

4.2.2. Real-time Text Conversation Languages

'dcsa' attributes can contain the SDP 'hlang-send' and 'hlang-recv'
attributes [RFC8373] to negotiate the language to be used for the
real-time text conversation.

For a T.140 data channel, the modality is "written" [RFC8373].

4.2.3. Real-time Text Direction

'dcsa' attributes can contain the SDP 'sendonly', 'recvonly',
'sendrecv' and 'inactive' attributes [I-D.ietf-mmusic-rfc4566bis] to
negotiate the direction in which text can be transmitted in a real-
time text conversation.

NOTE: A WebRTC data channel is always bi-directional. The usage of
the 'dcsa' attribute only affects the direction in which
implementations are allowed to transmit text on a T.140 data channel.

The offer/answer rules for the direction attributes are based on the
rules for unicast streams defined in [RFC3264], as described below.
Note that the rules only apply to the direction attributes.

Session level direction attributes [I-D.ietf-mmusic-rfc4566bis] have
no impact on a T.140 data channel.

4.2.3.1. Generating an Offer

If the offerer wishes to both send and receive text on a T.140 data
channel, it SHOULD mark the data channel as sendrecv with a
'sendrecv' attribute inside a 'dcsa' attribute. If the offerer does
not explicitly mark the data channel, an implicit 'sendrecv'
attribute inside a 'dcsa' attribute is applied by default.

If the offerer wishes to only send text on a T.140 data channel, it
MUST mark the data channel as sendonly with a 'sendonly' attribute
inside a 'dcsa' attribute.

If the offerer wishes to only receive text on a T.140 data channel, it MUST mark the data channel as recvonly with a 'recvonly' attribute inside a 'dcsa' attribute.

If the offerer wishes to neither send nor receive text on a T.140 data channel, it MUST mark the data channel as inactive with an 'inactive' attribute inside a 'dcsa' attribute.

If the offerer has marked a data channel as sendrecv (or if the offerer did not explicitly mark the data channel) or recvonly, it MUST be prepared to receive T.140 data as soon as the state of the T.140 data channel allows it.

4.2.3.2. Generating an Answer

When the answerer accepts an offer, and marks the direction of the text in the corresponding answer, the direction is based on the marking (or the lack of explicit marking) in the offer.

If the offerer explicitly marked the data channel as sendrecv, or if the offerer did not mark the data channel, the answerer SHOULD mark the data channel as sendrecv, sendonly, recvonly or inactive with a 'sendrecv', 'sendonly', 'recvonly' or 'inactive' attribute respectively inside a 'dcsa' attribute. If the answerer does not explicitly mark the data channel, an implicit 'sendrecv' attribute inside a 'dcsa' attribute is applied by default.

If the offerer marked the data channel as sendonly, the answerer MUST mark the data channel as recvonly or inactive with a 'recvonly' or 'inactive' attribute respectively inside a 'dcsa' attribute.

If the offerer marked the data channel as recvonly, the answerer MUST mark the data channel as sendonly or inactive with a 'sendonly' or 'inactive' attribute respectively inside a 'dcsa' attribute.

If the offerer marked the data channel as inactive, the answerer MUST mark the data channel as inactive with an 'inactive' attribute inside a 'dcsa' attribute.

If the answerer has marked a data channel as sendrecv or recvonly, it MUST be prepared to receive data as soon as the state of the T.140 data channel allows transmission of data.

4.2.3.3. Offerer Receiving an Answer

When the offerer receives an answer to the offer and the answerer has marked a data channel as sendrecv (or the answerer did not mark the data channel) or recvonly in the answer, the offerer can start

sending T.140 data as soon as the state of the T.140 data channel allows it. If the answerer has marked the data channel as inactive or sendonly, the offerer MUST NOT send any T.140 data.

If the answerer has not marked the direction of a T.140 data channel in accordance with the procedures above, it is RECOMMENDED that the offerer does not process that as an error situation, but rather assume that the answerer might both send and receive T.140 data on the data channel.

4.2.3.4. Modify Text Direction

If an endpoint wishes to modify a previously negotiated text direction in an ongoing session, it MUST initiate an offer that indicates the new direction, following the rules in Section 4.2.3.1. If the answerer accepts the offer it follows the procedures in Section 4.2.3.2.

4.3. Examples

Below is an example of an m= section of an offer for a T.140 data channel offering real-time text conversation in Spanish and Esperanto, and an m= section in the associated answer accepting Esperanto. The maximum character transmission rate is set to 20. As the offerer and answerer have not explicitly indicated the real-time text direction, the default direction "sendrecv" applies.

Offer:

```
m=application 911 UDP/DTLS/SCTP webrtc-datachannel
c=IN IP6 2001:db8::3
a=max-message-size:1000
a=sctp-port 5000
a=dcmap:2 label="ACME customer service";subprotocol="t140"
a=dcsa:2 fmtp:- cps=20
a=dcsa:2 hlang-send:es eo
a=dcsa:2 hlang-recv:es eo
```

Answer:

```
m=application 2004 UDP/DTLS/SCTP webrtc-datachannel
c=IN IP6 2001:db8::1
a=max-message-size:1000
a=sctp-port 6000
a=dcmap:2 label="ACME customer service";subprotocol="t140"
a=dcsa:2 fmtp:- cps=20
a=dcsa:2 hlang-send:eo
a=dcsa:2 hlang-recv:eo
```

Below is an example of an m= section of an offer for a T.140 data channel where the offerer wishes to only receive real-time text, and an m= section in the associated answer indicating that the answerer will only send real-time text. No maximum character transmission rate is indicated. No preference for the language to be used for the real-time text conversation is indicated.

   Offer:

      m=application 1400 UDP/DTLS/SCTP webrtc-datachannel
      c=IN IP6 2001:db8::3
      a=max-message-size:1000
      a=sctp-port 5000
      a=dcmap:2 label="ACME customer service";subprotocol="t140"
      a=dcsa:2 recvonly

   Answer:

      m=application 2400 UDP/DTLS/SCTP webrtc-datachannel
      c=IN IP6 2001:db8::1
      a=max-message-size:1000
      a=sctp-port 6000
      a=dcmap:2 label="ACME customer service";subprotocol="t140"
      a=dcsa:2 sendonly

5. T.140 Considerations

5.1. Session Layer Functions

   Section 6.1 of [T140] describes the generic T.140 session control
   functions at a high-level in a signalling protocol independent
   manner. The list below describes how the functions are realized when
   using a T.140 data channel.

   o Prepare session: An endpoint can indicate its support of T.140
      data channels using signalling specific means (e.g., using SIP
      OPTIONS [RFC3261]), or by indicating the support in an offer or
      answer (Section 4)
   o Initiate session: An offer used to request the establishment of a
      T.140 data channel (Section 4)
   o Accept session: An answer used to accept a request to establish a
      T.140 data channel (Section 4)
   o Deny session: An answer used to reject a request to establish a
      T.140 data channel, using the generic procedures for rejecting a
      data channel [I-D.ietf-mmusic-data-channel-sdpneg]
   o Disconnect session: An offer or answer used to disable a
      previously established T.140 data channel, using the generic
      procedures for closing a data channel
      [I-D.ietf-mmusic-data-channel-sdpneg]
   o Data: Data sent on an established T.140 data channel (Section 5.2)

5.2. Data Encoding and Sending

   T.140 text is encoded and framed as T140blocks [RFC4103].

   Each T140block is sent on the SCTP stream [RFC4960] used to realize
   the T.140 data channel using standard T.140 transmission procedures
   [T140]. One or more T140blocks can be sent in a single SCTP user
   message [RFC4960]. Unlike RTP based transport for real-time text
   [RFC4103], T.140 data channels do not use redundant transmission of
   text. The reason for this is that the T.140 data channel achieves
   robust transmission by using the "reliable" mode of the data channel.

   Data sending and reporting procedures conform to [T140].

   See Section 8 of [T140] for coding details.

   NOTE: The T.140 coding details contain information on optional
   control codes for controlling the presentation which may not be
   supported by the presentation level of the receiving application.
   The receiving application is expected to handle reception of such
   T.140 control codes appropriately (e.g. ignore and skip them) even if
   their effect on the presentation is not supported.

5.3. Data Buffering

   As described in [T140], buffering can be used to reduce overhead,
   with the maximum buffering time being 500 ms. It can also be used
   for staying within the maximum character transmission rate
   (Section 4.2), if such has been provided by the peer.

   An implementation needs to take the user requirements for smooth flow
   and low latency in real-time text conversation into consideration
   when assigning a buffer time. It is RECOMMENDED to use the default
   transmission interval of 300 milliseconds [RFC4103], or lower, for
   T.140 data channels.

5.4. Loss of T140blocks

   In case of network failure or congestion, T.140 data channels might
   fail and get torn down. If this happens but the session sustains, it
   is RECOMMENDED that implementations tries to reestablish the T.140
   data channels. If reestablishment of the T.140 data channel is
   successful, an implementation MUST evaluate if any T140blocks were
   lost. Retransmission of already successfully transmitted T140blocks
   MUST be avoided, and missing text markers [T140ad1] SHOULD be
   inserted in the received data stream where loss is detected or
   suspected.

NOTE: If the SCTP association [RFC4960] used to realize the T.140
data channel fails and gets torn down, it needs to be re-established
before the T.140 data channel can be reestablished. The procedure
after the reestablishment of the T.140 data channel defined in this
section apply no matter if only the T.140 data channel, or the whole
SCTP association, got torn down.

5.5. Multi-party Considerations

If an implementation needs to support multi-party scenarios, the
implementation needs to support multiple simultaneous T.140 data
channels, one for each remote party. At the time of writing this
document, this is true even in scenarios where each participant
communicates via a centralized conference server. The reason is
that, unlike RTP media, WebRTC data channels and the T.140 protocol
do not support the indication of the source of T.140 data. The SDP
'dcmap' attribute label attribute parameter (Section 4.1) can be used
by the offerer to provide additional information about each T.140
data channel, and help implementations to distinguish between them.

NOTE: Future extensions to T.140, or to the T140block, might allow
indicating the source of T.140 data, in which case it might be
possible to use a single T.140 data channel to transport data from
multiple remote sources. The usage of a single T.140 data channel,
without any protocol extensions, would require the conference server
to only forward real-time text from one source at any given time, and
e.g., include human readable text labels in the real-time text stream
that indicates the source whenever the conference server switches the
source. This would allow the receiver to present real-time text from
different sources separated. The procedures of such mechanism is
outside the scope of this document.

6. Gateway Considerations

A number of real-time text transports and protocols have been defined
for both packet-switched and circuit-switched networks. Many are
based on the ITU-T T.140 protocol on application and presentation
level [T140]. At the time of writing this document, some mechanisms
are no longer used, as the technologies they use have been obsoleted,
while others are still in use.

When performing interworking between T.140 data channels and real-
time text in other transports and protocols, a number of factors need
to be considered. At the time of writing this document, the most
common IP-based real-time text transport is the RTP based mechanism
defined in [RFC4103]. While this document does not define a complete
interworking solution, this list below provides some guidance and
considerations to take into account when designing a gateway for

interworking between T.140 data channels and RTP-based T.140
transport:

o For each T.140 data channel there is an RTP stream for real-time
   text [RFC4103]. Redundancy is by default declared and used on the
   RTP stream. There is no redundancy on the T.140 data channel, but
   the reliable property [I-D.ietf-mmusic-data-channel-sdpneg] is set
   on it.

o During a normal text flow, T140blocks received from one network
   are forwarded towards the other network. Keep-alive traffic is
   implicit on the T.140 data channel. A gateway might have to
   extract keep-alives from incoming RTP streams, and MAY generate
   keep-alives on outgoing RTP streams.

o If the gateway detects or suspects loss of data on the RTP stream,
   and the lost data has not been retrieved using a redundancy
   mechanism, the gateway SHOULD insert the T.140 missing text marker
   [T140ad1] in the data sent on the outgoing T.140 data channel.

o If the gateway detects that the T.140 data channel has failed and
   got torn down, once the data channel has been reestablished the
   gateway SHOULD insert the T.140 missing text marker [T140ad1] in
   the data sent on the outgoing RTP stream if it detects or suspects
   that data sent by the remote T.140 data channel endpoint was lost.

o If the gateway detects that the T.140 data channel has failed and
   got torn down, once the data channel has been reestablished the
   gateway SHOULD insert the T.140 missing text marker [T140ad1] in
   the data sent on the outgoing T.140 data channel if it detects or
   suspects that data sent or to be sent on the T.140 data channel
   was lost during the failure.

o The gateway MUST indicate the same text transmission direction
   (Section 4.2.3) on the T.140 data channel and the RTP stream.

NOTE: In order for the gateway to insert a missing text marker, or to
perform other actions that require that the gateway has access to the
T.140 data, the T.140 data cannot be encrypted end-to-end between the
T.140 data channel endpoint and the RTP endpoint. At the time of
writing this document, no mechanism to provide such end-to-end
encryption is defined.

7. Update to RFC 8373

This document updates RFC 8373 [RFC8373], by defining how the SDP
hlang-send and hlang-recv attributes are used for the "application/
webrtc-datachannel" media type.

SDP offerers and answerers MUST NOT include the attributes directly
in the m= section associated with the 'application/webrtc-
datachannel' media type. Instead, the attributes MUST be associated
with individual data channels, using the SDP 'dcsa' attribute. A

specification that defines a subprotocol that uses the attributes
MUST specify the modality for that subprotocol, or how to retrieve
the modality if the subprotocol supports multiple modalities. The
subprotocol is indicated using the SDP 'dcmap' attribute.

8. Security Considerations

The generic security considerations for WebRTC data channels are
defined in [I-D.ietf-rtcweb-data-channel]. As data channels are
always encrypted by design, the T.140 data channels will also be
encrypted.

The generic security considerations for the SDP-based external
negotiation method are defined in
[I-D.ietf-mmusic-data-channel-sdpneg]. There are no additional T.140
data channel specific security considerations.

9. IANA considerations

[RFC EDITOR NOTE: Please replace all instances of RFCXXXX with the
RFC number of this document.]

9.1. Subprotocol Identifier t140

This document adds the subprotocol identifier "t140" to the
"WebSocket Subprotocol Name Registry" as follows:

```
              +---------------------------+-------------+
              | Subprotocol Identifier:   | t140        |
              | Subprotocol Common Name:  | ITU-T T.140 |
              | Subprotocol Definition:   | RFCXXXX     |
              | Reference:                | RFCXXXX     |
              +---------------------------+-------------+
```

9.2. SDP fmtp Attribute

This document modifies the usage of the SDP 'fmtp' attribute, if this
attribute is included in an SDP 'dcsa' attribute and associated with
an T.140 real-time text session over a WebRTC data channel. The
modified usage is described in Section 4.2.1.

The usage level "dcsa(t140)" is added to the IANA registration of the
SDP 'fmtp' attribute as follows:

```
+-----------------------+-------------------------------------+
| Contact name:         | IESG                                |
| Contact email:        | iesg@ietf.org                       |
| Attribute name:       | fmtp                                |
| Usage level:          | dcsa(t140)                          |
| Purpose:              | Indicate the maximum transmission rate |
|                       | that an endpoint is willing to receive on |
|                       | a T.140 data channel.               |
| Reference:            | RFCXXXX                             |
+-----------------------+-------------------------------------+
```

9.3. SDP Language Attributes

This document modifies the usage of the SDP 'hlang-send' and 'hlang-
recv' attributes, if these attributes are included in SDP 'dcsa'
attributes associated with an T.140 data channel. The modified usage
is described in Section 4.2.2.

The usage level "dcsa(t140)" is added to the IANA registration of the
SDP 'hlang-send' attribute as follows:

```
+-----------------------+-------------------------------------+
| Contact name:         | IESG                                |
| Contact email:        | iesg@ietf.org                       |
| Attribute name:       | hlang-send                          |
| Usage level:          | dcsa(t140)                          |
| Purpose:              | Negotiate the language to be used on a |
|                       | T.140 data channel.                 |
| Reference:            | RFCXXXX                             |
+-----------------------+-------------------------------------+
```

The usage level "dcsa(t140)" is added to the IANA registration of the
SDP 'hlang-recv' attribute as follows:

```
+-----------------------+-------------------------------------+
| Contact name:         | IESG                                |
| Contact email:        | iesg@ietf.org                       |
| Attribute name:       | hlang-recv                          |
| Usage level:          | dcsa(t140)                          |
| Purpose:              | Negotiate the language to be used on a |
|                       | T.140 data channel.                 |
| Reference:            | RFCXXXX                             |
+-----------------------+-------------------------------------+
```

9.4. SDP Media Direction Attributes

   This document modifies the usage of the SDP 'sendonly', 'recvonly',
   'sendrecv' and 'inactive' attributes, if these attributes are
   included in SDP 'dcsa' attributes associated T.140 data channel. The
   modified usage is described in Section 4.2.3.

   The usage level "dcsa(t140)" is added to the IANA registration of the
   SDP 'sendonly' attribute as follows:

   +-----------------------+---+
Contact name:	IESG
Contact email:	iesg@ietf.org
Attribute name:	sendonly
Usage level:	dcsa(t140)
Purpose:	Negotiate the direction in which real-
	time text can be sent on a T.140 data
	channel.
Reference:	RFCXXXX
   +-----------------------+---+

   The usage level "dcsa(t140)" is added to the IANA registration of the
   SDP 'recvonly' attribute as follows:

   +-----------------------+---+
Contact name:	IESG
Contact email:	iesg@ietf.org
Attribute name:	recvonly
Usage level:	dcsa(t140)
Purpose:	Negotiate the direction in which real-
	time text can be sent on a T.140 data
	channel.
Reference:	RFCXXXX
   +-----------------------+---+

   The usage level "dcsa(t140)" is added to the IANA registration of the
   SDP 'sendrecv' attribute as follows:

   +-----------------------+---+
Contact name:	IESG
Contact email:	iesg@ietf.org
Attribute name:	sendrecv
Usage level:	dcsa(t140)
Purpose:	Negotiate the direction in which real-
	time text can be sent on a T.140 data
	channel.
Reference:	RFCXXXX
   +-----------------------+---+

The usage level "dcsa(t140)" is added to the IANA registration of the
SDP 'inactive' attribute as follows:

```
+-----------------------+-------------------------------------------+
| Contact name:         | IESG                                      |
| Contact email:        | iesg@ietf.org                             |
| Attribute name:       | inactive                                  |
| Usage level:          | dcsa(t140)                                |
| Purpose:              | Negotiate the direction in which real-    |
|                       | time text can be sent on a T.140 data     |
|                       | channel.                                  |
| Reference:            | RFCXXXX                                   |
+-----------------------+-------------------------------------------+
```

10. Acknowledgements

This document is based on an earlier Internet draft edited by Keith
Drage, Juergen Stoetzer-Bradler and Albrecht Schwarz.

Thomas Belling provided useful comments on the initial (pre-
submission) version of the draft. Gunnar Hellstrom provided comments
and text on the draft. Paul Kyzivat and Bernard Aboba provided
comments on the draft.

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

   [RFC4103] Hellstrom, G. and P. Jones, "RTP Payload for Text
              Conversation", RFC 4103, DOI 10.17487/RFC4103, June 2005,
              <https://www.rfc-editor.org/info/rfc4103>.

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

   [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>.

   [RFC8373] Gellens, R., "Negotiating Human Language in Real-Time
              Communications", RFC 8373, DOI 10.17487/RFC8373, May 2018,
              <https://www.rfc-editor.org/info/rfc8373>.

   [I-D.ietf-mmusic-rfc4566bis]
              Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP:
              Session Description Protocol", draft-ietf-mmusic-
              rfc4566bis-37 (work in progress), August 2019.

   [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-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.

   [T140]   ITU-T, "Recommendation ITU-T T.140 (02/1998), Protocol for
              multimedia application text conversation", February 1998.

   [T140ad1] ITU-T, "Recommendation ITU-T.140 Addendum 1 - (02/2000),
              Protocol for multimedia application text conversation",
              February 2000.

11.2. Informative References

   [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>.

   [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>.

   [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.

Author's Address

    Christer Holmberg
    Ericsson
    Hirsalantie 11
    Jorvas 02420
    Finland

    Email: christer.holmberg@ericsson.com