MMUSIC Working Group C. Holmberg
Internet-Draft Ericsson
Updates: RFC8373 (if approved) August 29, 2019
Intended status: Standards Track
Expires: March 1, 2020

T.140 Real-time Text Conversation over WebRTC Data Channels
draft-holmberg-mmusic-t140-usage-data-channel-01

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.

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 native transport for IP networks is the "RTP Payload for Text Conversation" [RFC4103] mechanism, based on the Real-time Transport Protocol (RTP) [RFC4103].

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 1: This WebRTC term of a "T.140 data channel" is actually synonym to the originally introduced concept of a "T.140 data channel" for the T.140 protocol, see Section 4.3 of [T140].

NOTE 2: The decision to transport realtime text over a data channel, instead of using RTP based transport [RFC4103], in WebRTC is constituted by use-case "U-C 5: Realtime 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 synonym for the text conversation protocol according to [T140].

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. SDP Considerations

The generic SDP considerations, including the SDP Offer/Answer proceudres, 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.

3.1. 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 descripton (m= section) [RFC4566] 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 negotied 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 also to data channels negotiated using the mechanism defined in this document.

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

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

```
a=dcmap:3 subprotocol="t140"
```

3.2. Use of dcsa Attribute

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

3.2.1. Maximum Character Transmission

A 'dcsa' attribute can contain the SDP fmtp attribute used to indicate a maximum character transmission rate [RFC4103]. The 'cps' attribute parameter is used indicate the maximum character transmission rate that the endpoint that includes the attribute is able to receive. The 'format' attribute parameter is not used with T.140 data channels, and MUST be set to "-".

If the fmtp attribute is not included, it indicates that no maximum character transmission rate is indicated. It does not mean that the default value of 30 applies [RFC4103].

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

NOTE: The 'cps' attribute parameter is especially useful when a T.140 data channel endpoint is acting as a gateway [Section 5] and is interworking with a T.140 transport mechanism that have restrictions on how many characters can be sent per second.

3.2.2. 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].

3.3. Examples

Below is an example of an m= section describing a T.140 data channel, where the maximum character transmission rate is set to 20.

```
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:1 label="text conversation";subprotocol="t140"
a=dcsa:1 fmtp:- cps=20
```

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.

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:1 label="ACME customer service";subprotocol="t140"
a=dcsa:1 fmtp:- cps=30
a=dcsa:1 hlang-send:es eo
a=dcsa:1 hlang-recv:es eo
```

Answer:

```
m=application 911 UDP/DTLS/SCTP webrtc-datachannel
c=IN IP6 2001:db8::1
a=max-message-size:1000
a=sctp-port 5000
a=dcmap:1 label="ACME customer service";subprotocol="t140"
a=dcsa:1 fmtp:- cps=30
a=dcsa:1 hlang-send:eo
a=dcsa:1 hlang-recv:eo
```

4. T.140 Considerations

4.1. Session Layer Functions

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

- 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 3)
- Initiate session: An offer used to request the establishment of a T.140 data channel (Section 3)
- Accept session: An answer used to accept a request to establish a T.140 data channel (Section 3)
- 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]
- 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]
- Data: Data sent on an established T.140 data channel (Section 4.2)

4.2. 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 realtime text [RFC4103], T.140 data channels do not use redundant transmission of text.

Data sending and reporting procedures conform to [T140].

See Section 8 of [T140] for coding details.

4.3. 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 3.2), 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, also for T.140 data channels.

4.4. Loss of T140blocks

In case of network failure or congestion, T140 data channels might fail and get torn down. If this happens but the session sustains, it is RECOMMENDED that a low number of retries are made to reestablish the T140 data channels. If reestablishment of the T140 data channel is successful, an implementation MUST evaluate if any T140blocks were lost. Retransmission of already transmitted T140blocks MUST be avoided, and missing text markers [T140ad1] SHOULD be inserted in the received data stream where loss is detected or suspected.

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, also for T.140 data channels.

4.5. 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 participants communicate 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 3.1) 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.

5. 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 etc, while others are still in use.

When performing interworking between T.140 data channels and another real-time text transports and protocols with real-time text in another, 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 an gateway for interworking between T140 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 RTP stream. On the T.140 data channel there is no redundancy, but the reliable property [I-D.ietf-mmusic-data-channel-sdpneg] of T.140 the data channel is set.
- 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 It is RECOMMENDED that the gateway uses the same transmission interval on both the T140 data channel and the RTP stream, if possible. That will reduce the delay caused by buffering.
- o If the gateway detects or suspects loss of data on the RTP stream, the gateway 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 or suspects loss of data on the T.140 data channel, the gateway gateway SHOULD insert the T.140 missing text marker [T140ad1] in the data sent on the outgoing RTP stream.
- 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.

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, a mechanism to provide such end-to-end encryption has not been defined.

6. Update to RFC 8373

This document updates RFC 8373, 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 retreive the modality if the subprotocol supports multiple modalities.

7. 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].

8. IANA considerations

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

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. 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.

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, <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>.

[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>.

[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-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 aEUR" Addendum 1 (02/2000), "Protocol for multimedia application text conversation"", February 2000.

10.2. 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>.

[V151] ITU-T, "Recommendation ITU-T V.151 (05/2006), "Procedures for the end-to-end connection of analogue PSTN text telephones over an IP network utilizing text relay"", May 2006.

[V152] ITU-T, "Recommendation ITU-T V.152 (09/2010), "Procedures for supporting voice-band data over IP networks"", September 2010.

[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