

Workgroup: AVTCore  
Internet-Draft:  
draft-ietf-avtcore-multi-party-rtt-mix-02  
Updates: [RFC 4102](#), [RFC 4103](#) (if approved)  
Published: 20 May 2020  
Intended Status: Standards Track  
Expires: 21 November 2020  
Authors: G. Hellstrom  
Gunnar Hellstrom Accessible Communication  
**RTP-mixer formatting of multi-party Real-time text**

## Abstract

Real-time text mixers for multi-party sessions need to identify the source of each transmitted group of text so that the text can be presented by endpoints in suitable grouping with other text from the same source.

Regional regulatory requirements specify provision of real-time text in multi-party calls. RFC 4103 mixer implementations can use traditional RTP functions for source identification, but the mixer source switching performance is limited when using the default transmission with redundancy.

An enhancement for RFC 4103 real-time text mixing is provided in the present specification, suitable for a centralized conference model that enables source identification and efficient source switching. The intended use is for real-time text mixers and multi-party-aware participant endpoints. The mechanism builds on use of the CSRC list in the RTP packet and an extended packet format 'text/rex'.

A capability exchange is specified so that it can be verified that a participant can handle the multi-party coded real-time text stream. The capability is indicated by the media subtype "text/rex".

The document updates RFC 4102[[RFC4102](#)] and RFC 4103[[RFC4103](#)]

A brief description about how a mixer can format text for the case when the endpoint is not multi-party aware is also provided.

## Status of This Memo

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

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

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

This Internet-Draft will expire on 21 November 2020.

## Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

- [1. Introduction](#)
  - [1.1. Selected solution and considered alternative](#)
  - [1.2. Nomenclature](#)
  - [1.3. Intended application](#)
- [2. Use of fields in the RTP packets](#)
- [3. Actions at transmission by a mixer](#)
- [4. Actions at reception](#)
- [5. RTCP considerations](#)
- [6. Chained operation](#)
- [7. Usage without redundancy](#)
- [8. Use with SIP centralized conferencing framework](#)
- [9. Media Subtype Registration](#)

## [10. SDP considerations](#)

### [10.1. Security for session control and media](#)

### [10.2. SDP offer/answer examples](#)

## [11. Examples](#)

## [12. Performance considerations](#)

## [13. Presentation level considerations](#)

### [13.1. Presentation by multi-party aware endpoints](#)

### [13.2. Multi-party mixing for multi-party unaware endpoints](#)

## [14. Gateway Considerations](#)

### [14.1. Gateway considerations with Textphones \(e.g. TTYs\).](#)

### [14.2. Gateway considerations with WebRTC.](#)

## [15. Updates to RFC 4102 and RFC 4103](#)

## [16. Congestion considerations](#)

## [17. Acknowledgements](#)

## [18. IANA Considerations](#)

## [19. Security Considerations](#)

## [20. Change history](#)

### [20.1. Changes included in draft-ietf-avtc core-multi-party-rtt-mix-02](#)

### [20.2. Changes to draft-ietf-avtc core-multi-party-rtt-mix-01](#)

### [20.3. Changes from draft-hellstrom-avtc core-multi-party-rtt-source-03 to draft-ietf-avtc core-multi-party-rtt-mix-00](#)

### [20.4. Changes from draft-hellstrom-avtc core-multi-party-rtt-source-02 to -03](#)

### [20.5. Changes from draft-hellstrom-avtc core-multi-party-rtt-source-01 to -02](#)

### [20.6. Changes from draft-hellstrom-avtc core-multi-party-rtt-source-00 to -01](#)

## [21. References](#)

### [21.1. Normative References](#)

### [21.2. Informative References](#)

## [Author's Address](#)

## **1. Introduction**

RFC 4103 [[RFC4103](#)] specifies use of RFC 3550 RTP [[RFC3550](#)] for transmission of real-time text (RTT) and the "text/t140" format. It also specifies a redundancy format "text/red" for increased robustness. RFC 4102 [[RFC4102](#)] registers the "text/red" format. Regional regulatory requirements specify provision of real-time text in multi-party calls.

The redundancy scheme enables efficient transmission of redundant text in packets together with new text. However the redundant header format has no source indicators for the redundant transmissions. An assumption has had to be made that the redundant parts in a packet are from the same source as the new text. The recommended transmission is one new and two redundant generations of text (T140blocks) in each packet and the recommended transmission interval is 300 ms.

A mixer, selecting between text input from different sources and transmitting it in a common stream needs to make sure that the receiver can assign the received text to the proper sources for presentation. Therefore, using RFC 4103 without any extra rule for source identification, the mixer needs to stop sending new text from that source and then make sure that all text so far has been sent with all intended redundancy levels (usually two) before switching source. That causes the very long time of one second to switch between transmission of text from one source to text from another source. Both the total throughput and the switching performance in the mixer is too low for most applications.

A more efficient source identification scheme requires that each redundant T140block has its source individually preserved. The present specification introduces a source indicator by specific rules for populating the CSRC-list and the redundancy header in the RTP-packet.

An extended packet format 'text/rex' is specified for this purpose, providing the possibility to include text from up to 16 sources in each packet in order to enhance mixer source switching performance.

A negotiation mechanism can therefore be based on selection between the "text/red" and the "text/rex" media formats for verification that the receiver is able to handle the multi-party coded stream.

A fall-back mixing procedure is specified for cases when the negotiation results in "text/red" being the only common submedia format.

The document updates RFC 4102[[RFC4102](#)] and RFC 4103[[RFC4103](#)] by introducing an extended packet format for the multi-party mixing case and more strict rules for the use of redundancy.

### **1.1. Selected solution and considered alternative**

The mechanism specified in the present document makes use of the RTP mixer model specified in RFC3550[[RFC3550](#)]. From some points of view, use of the RTP translator model specified in RFC 3550 would be more efficient, because then the text packets can pass the translator with only minor modification. However, there may be a lack of support for the translator model in existing RTP implementations, and therefore the more common RTP-mixer model was selected.

### **1.2. Nomenclature**

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](#)].

The terms SDES, CNAME, NAME, SSRC, CSRC, CSRC list, CC are explained in [[RFC3550](#)]

The term "T140block" is defined in RFC 4103 [[RFC4103](#)] to contain one or more T.140 code elements.

### **1.3. Intended application**

The scheme for identification of source of redundant transmissions is intended for transmission from entities taking the mixer role in centralised mixing configurations for RTT. It is intended for reception by both endpoints and mixers.

## **2. Use of fields in the RTP packets**

RFC 4103[[RFC4103](#)] specifies use of RFC 3550 RTP[[RFC3550](#)], and a redundancy format "text/red" for increased robustness of real-time text transmission. The current specification updates RFC 4102[[RFC4102](#)] and RFC 4103[[RFC4103](#)] by introducing a rule for populating and using the CSRC-list in the RTP packet and extending the redundancy header in order to enhance the performance in multi-party RTT sessions.

The "text/rex" format is an "n-tuple" of the "text/red" format intended to carry text information from up to 16 sources per packet. The CSRC members, the redundancy header and the T140block areas SHALL be populated with a number of the same fields as they are specified in RFC 4103[RFC4103] appended in the same order as the sources appear in the CSRC list. The most recent text should be placed as the last text element.

When transmitted from a mixer, the members in the CSRC-list SHALL contain the SSRCs of the sources of the T140blocks in the packet. The order of the CSRC members MUST be the same as the order of sources of the T140blocks and redundant header fields. When redundancy is used, text from all included sources MUST have the same number of redundancy generations. The primary, first redundant, second redundant and possible further redundant generations of T140blocks MUST be grouped per source in the packet in "source groups". The recommended level of redundancy is to use one primary and two redundant generations of T140blocks. In some cases, a primary or redundant T140block is empty, but is still represented by a member in the redundancy header.

The CC field SHALL show the number of members in the CSRC list.

The RTP header is followed by one or more source groups of data block headers: one header for each text block to be included. Each of these headers except the last provides the timestamp offset and length of the corresponding data block, in addition to the payload type number corresponding to the payload format "text/t140". The data block headers are followed by the data fields carrying T140blocks from the sources. Finally, the most recent T140block (if any) follows.

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
+--+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+--+									
F       block PT										timestamp offset																				block length									
+--+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+--+										+--+--+--+--+--+--+--+--+--+									

The bits in the header are specified as follows:

**F:** 1 bit First bit in header indicates whether another header block follows. It has value 1 if further header blocks follow, and value 0 if this is the last header block.

**block PT:** 7 bits RTP payload type number for this block, corresponding to the t140 payload type.

**timestamp offset:** 14 bits Unsigned offset of timestamp of this block relative to timestamp given in RTP header. The offset is a time to be subtracted from the current timestamp to determine the

timestamp of the data when this block was received to the mixer. For primary data from a mixer, the resulting time is the time when the data or the youngest part of the data was received by the mixer. For primary data from an endpoint, only one primary data is included last in the packet. If the timestamp offset would be >15 000, it is set to 15 000. For redundant data, the resulting time the resulting time is the time when the data was sent as primary. If the value would be >15 000, then it SHALL be set to 15 000 plus 300 times the redundancy level of the data. The high values appear only in exceptional cases, e.g. when some data has been held in order to keep the text flow under the cps limit.

**block length:** 10 bits Length in bytes of the corresponding data block excluding header.

The last data block is placed last in the packet. The timestamp and block-length fields are derived from the header of this block, since they may be determined from the RTP header and overall packet length. The header for the final block comprises only a zero F bit, and the block payload type number information corresponding to the t140 format, a total of 8 bits. This is illustrated in the figure below:

```

0 1 2 3 4 5 6 7
+--+--+--+--+--+
|0|   Block PT   |
+--+--+--+--+--+

```

This specification departs from section 4 of RFC 2198 [[RFC2198](#)] which associates the whole of the CSRC-list with the primary data and assumes that the same list applies to reconstructed redundant data. In the present specification a T140block is associated with exactly one CSRC list member as described above. Also RFC 2198 [[RFC2198](#)] anticipates infrequent change to CSRCs; implementers should be aware that the order of the CSRC-list according to this specification will vary during transitions between transmission from the mixer of text originated by different participants.

The picture below shows a typical 'text/rex' RTP packet with multi-party RTT contents and coding according to the present specification.

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|V=2|P|X| CC=3 |M|  "REX" PT   |   RTP sequence number           |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                               timestamp of packet creation        |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                               synchronization source (SSRC) identifier
+=====+
|  CSRC list member 1 = SSRC of source of "A"                      |
|  CSRC list member 2 = SSRC of source of "B"                      |
|  CSRC list member 3 = SSRC of source of "C"                      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|1|   T140 PT   |timestamp offset of "A-R2" |"A-R2" block length|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|1|   T140 PT   |timestamp offset of "A-R1" |"A-R1" block length|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|1|   T140 PT   | timestamp offset of "A-P" |"A-P" block length |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|1|   T140 PT   |timestamp offset of "B-R2" |"B-R2" block length|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|1|   T140 PT   |timestamp offset of "B-R1" |"B-R1" block length|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|1|   T140 PT   | timestamp offset of "B-P" | "B-P" block length|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|1|   T140 PT   |timestamp offset of "C-R2" |"C-R2" block length|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|1|   T140 PT   |timestamp offset of "C-R1" |"C-R1" block length|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|0|   T140 PT   | "A-R2" T.140 encoded redundant data             |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+-----+
|  |  "A-R1" T.140 encoded redundant data  |                       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+-----+
|  "A-P" T.140 encoded primary data  |                               |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+-----+
|      "B-R2" T.140 encoded redundant data      |                   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+-----+
|  |  "B-R1" T.140 encoded redundant data  |                       |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+-----+
|  "B-P" T.140 encoded primary data              |                   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+-----+
|      "C-R2" T.140 encoded redundant data      |                   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+-----+
|  |  "C-R1" T.140 encoded redundant data      |                   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+-----+
|      "C-P" T.140 encoded primary data              |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Figure 1:A 'text/rex' packet with text from three sources A, B, C.



### 3. Actions at transmission by a mixer

A "text/rex" transmitter is sending packets at least at regular transmission intervals as long as there is something (new or redundant T140blocks) to transmit. The default transmission interval that SHOULD be applied is 300 ms.

As soon as a participant is known to participate in a session and being available for text reception, a Unicode BOM character SHALL be sent to it according to the procedures in the present document.

After that, the transmitter SHALL send keep-alive traffic to the receivers at regular intervals when no other traffic has occurred during that interval if that is decided for the actual connection. Recommendations for keep-alive can be found in RFC 6263[\[RFC6263\]](#).

For multi-party operation, it is RECOMMENDED that the mixer sends a packet to each receiver as soon as text has been received from a source as long as the maximum number of characters per second indicated by the recipient is not exceeded, and also the number of packets sent per second to a recipient is kept under a specified number. This number SHALL be 10 if no other limit is applied for the application. The intention is to keep the latency introduced by the mixer low.

The mixer has its own SSRC, and its own RTP sequence number series.

The number of redundant generations of T140blocks to include in transmitted packets SHALL be deducted from the SDP negotiation. It SHOULD be set to the minimum of the number declared by the receiver and the transmitter. The same number of redundant generations MUST be used for all sources in the transmissions. The number of generations sent to a receiver SHALL be the same during the whole session unless it is modified by session renegotiation.

At time of transmission, the mixer SHALL populate the RTP packet with T140blocks combined from all T140blocks queued for transmission originating from each source as long as this is not in conflict with the allowed number of characters per second or the maximum packet size. These T140blocks SHALL be placed in the packet interleaved with redundant T140blocks and new T140blocks from other sources. The primary and redundant T140blocks from each source are grouped together in "source groups" appearing in the packet in the same order as the corresponding CSRC list member and redundancy header members for the source group. The format of the redundancy header members and the T140blocks in each source group are the same as for the single source possible in the "text/red" format. The SSRC of each source shall be placed as a member in the CSRC-list at a place corresponding to the place of its T140blocks in the packet. The

current time SHALL be inserted in the timestamp. The timestamp offset values for empty T140blocks have no relevance but SHOULD be assigned realistic values.

Text from a maximum of 16 sources MAY be included in a packet. The reason for this limitation is the maximum number of CSRC list members. If text from more sources need to be transmitted, the mixer MAY let the sources take turns in having their text transmitted. When stopping transmission of one source to allow another source to have its text sent, all intended redundant generations of the last text from the source to be stopped MUST be transmitted before text from another source can be transmitted. Actively transmitting sources SHOULD be allowed to take turns with short intervals to have their text transmitted.

If no unsent T140blocks were available at this time for a source, but T140blocks are available which have not yet been sent the full intended number of redundant transmissions, then the primary T140block for that source is composed of an empty T140block, and populated (without taking up any length) in a packet for transmission. The corresponding SSRC SHALL be placed in its place in the CSRC-list.

The primary T140block from each source in the latest transmission is used to populate the first redundant T140block for that source. The first redundant T140block for that source from the latest transmission is placed as the second redundant T140block source.

Usually this is the level of redundancy used. If a higher number of redundancy is negotiated, then the procedure SHALL be maintained until all available redundant levels of T140blocks and their sources are placed in the packet. If a receiver has negotiated a lower number of text/rex generations, then that level shall be the maximum used by the transmitter.

The timer offset values are inserted in the redundancy header, with the time offset from the RTP timestamp in the packet when the corresponding T140block was sent from the mixer as primary.

The number of members in the CSRC list shall be placed in the "CC" header field. Only mixers place values >0 in the "CC" field.

When there is no new T140block to transmit, and no redundant T140block that has not been retransmitted the intended number of times, the transmission process can stop until either new T140blocks arrive, or a keep-alive method calls for transmission of keep-alive packets.

#### 4. Actions at reception

The "text/rex" receiver included in an endpoint with presentation functions will receive RTP packets in the single stream from the mixer, and SHALL distribute the T140blocks for presentation in presentation areas for each source. Other receiver roles, such as gateways or chained mixers are also feasible, and requires consideration if the stream shall just be forwarded, or distributed based on the different sources.

If the "CC" field value of a received packet is  $>1$ , it indicates that multi-party transmission is active, and the receiver MUST act on the different sources according to its role. If the CC field value is 1, it indicates that there is a mixer between the source and the receiver, but that there is only one source at the moment. If the CC value is 0, the connection is point-to-point.

The used level of redundancy generations SHALL be evaluated from the received packet contents. If the CC value is 0, the number of generations (including the primary) is equal to the number of members in the redundancy header. If the CC value is  $>0$ , the number of generations (including the primary) is equal to the number of members in the redundancy header divided by the CC value. If the remainder from the division is  $>0$ , then the packet is malformed and SHALL cause an error indication in the receiver.

The RTP sequence numbers of the received packets SHALL be monitored for gaps and packets out of order.

As long as the sequence is correct, each packet SHALL be unpacked in order. The T140blocks SHALL be extracted from the primary areas, and the corresponding SSRCS SHALL be extracted from the corresponding positions in the CSRC list and used for assigning the new T140block to the correct presentation areas (or correspondingly).

If a sequence number gap appears and is still there after some defined time for jitter resolution, T140data SHALL be recovered from redundant data. If the gap is wider than the number of generations of redundant T140blocks in the packet, then a t140block SHALL be created with a marker for text loss [[T140ad1](#)] and assigned to the SSRC of the transmitter as a general input from the mixer because in general it is not possible to deduct from which sources text was lost.

Then, the T140blocks in the received packet SHALL be retrieved beginning with the highest redundant generation, grouping them with the corresponding SSRC from the CSRC-list and assigning them to the presentation areas per source. Finally the primary T140blocks SHALL be retrieved from the packet and similarly their sources retrieved

from the corresponding positions in the CSRC-list, and then assigned to the corresponding presentation areas for the sources.

If the sequence number gap was equal to or less than the number of redundancy generations in the received packet, a missing text marker SHALL NOT be inserted, and instead the T140blocks and their SSRCS fully recovered from the redundancy information and the CSRC-list in the way indicated above.

Unicode character BOM is used as a start indication and sometimes used as a filler or keep alive by transmission implementations. These SHALL be deleted on reception.

Empty T140blocks are included as fillers for unused redundancy levels in the packets. They just do not provide any contents and do not contribute to the received streams.

## **5. RTCP considerations**

A mixer SHALL send RTCP reports with SDES, CNAME and NAME information about the sources in the multi-party call. This makes it possible for participants to compose a suitable label for text from each source.

## **6. Chained operation**

By strictly applying the rules for "text/rex" packet format by all conforming devices, mixers MAY be arranged in chains.

## **7. Usage without redundancy**

The "text/rex" format SHALL be used also for multi-party communication when the redundancy mechanism is not used. That MAY be the case when robustness in transmission is provided by some other means than by redundancy. All aspects of the present document SHALL be applied except the redundant generations in transmission.

The "text/rex" format SHOULD thus be used for multi-party operation, also when some other protection against packet loss is utilized, for example a reliable network or transport. The format is also allowed to be used for point-to-point operation.

## **8. Use with SIP centralized conferencing framework**

The SIP conferencing framework, mainly specified in RFC 4353[RFC4353], RFC 4579[RFC4579] and RFC 4575[RFC4575] is suitable for coordinating sessions including multi-party RTT. The RTT stream between the mixer and a participant is one and the same during the conference. Participants get announced by notifications when participants are joining or leaving, and further user information

may be provided. The SSRC of the text to expect from joined users MAY be included in a notification. The notifications MAY be used both for security purposes and for translation to a label for presentation to other users.

Note: The CSRC-list in an RTP packet only includes participants who's text is included in one or more text blocks. It is not the same as the total list of participants in a conference. With audio and video media, the CSRC-list would often contain all participants who are not muted whereas text participants that don't type are completely silent and thus are not represented in RTP packet CSRC-lists.

## 9. Media Subtype Registration

This registration is done using the template defined in [[RFC6838](#)] and following [[RFC4855](#)].

**Type name:** text

**Subtype name:** rex

**Required parameters:**

**rate:** The RTP timestamp (clock) rate. The only valid value is 1000.

**pt:** A slash-separated list with the payload type number(pt) for the primary text, the first redundant text, the second redundant text etc, that the receiver is capable to receive.

**Optional parameter: cps:** This parameter is used to signal the capabilities of a receiver implementation. It indicates the maximum number of characters that may be received per second measured over a period of 10 seconds. The default value is 150.

**Encoding considerations:** binary; see Section 4.8 of [[RFC6838](#)].

**Security considerations:** See [Section 19](#) of RFC xxxx. [RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

**Interoperability considerations:** None.

**Published specification:** RFC XXXX. [RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

**Applications which use this media type:** For example: Text conferencing tools, multimedia conferencing tools.

**Fragment identifier considerations:**

N/A.

**Additional information:** None.

**Person & email address to contact for further information:** Gunnar Hellstrom <gunnar.hellstrom@ghaccess.se>

**Intended usage:** COMMON

**Restrictions on usage:** This media type depends on RTP framing, and hence is only defined for transfer via RTP [[RFC3550](#)].

**Author:** Gunnar Hellstrom <gunnar.hellstrom@ghaccess.se>

**Change controller:** IETF AVTCore Working Group delegated from the IESG.

## **10. SDP considerations**

There are receiving RTT implementations which implement RFC 4103 [[RFC4103](#)] but not the source separation by the CSRC. Sending mixed text according to the usual CSRC convention from RFC 2198 [[RFC2198](#)] to a device implementing only RFC 4103 [[RFC4103](#)] would risk to lead to unreadable presented text. Therefore, in order to negotiate RTT mixing capability according to the present specification, all devices supporting the present specification for multi-party aware participants SHALL include an sdp media format "text/rex" in the sdp, indicating this capability in offers and answers. Multi-party streams using the coding of the present specification intended for multi-party aware endpoints MUST NOT be sent to devices which have not indicated the "text/rex" format.

Implementations not understanding this format MUST ignore it according to common SDP rules.

The sdp media format defined here, is named "rex", for extended redundancy. It is intended to be used in "text" media descriptions with "text/rex" and "text/t140" formats. Both formats MUST be declared for the "text/rex" format to be used. It indicates capability to use source indications in the CSRC list and the packet format according to the present specification. It also indicates ability to receive 150 real-time text characters per second.

### **10.1. Security for session control and media**

Security SHOULD be applied on both session control and media. In applications where legacy endpoints without security may exist, a negotiation between security and no security SHOULD be applied. If no other security solution is mandated by the application, then RFC

8643 OSRTP[[RFC8643](#)] SHOULD be applied to negotiate SRTP media security with DTLS. Most SDP examples below are expressed without the security additions for simplicity. The principles (but not all details) for applying DTLS-SRTP security is shown in a couple of the following examples.

## 10.2. SDP offer/answer examples

Offer example for just multi-party capability:

```
m=text 11000 RTP/AVP 101 98
a=rtpmap:98 t140/1000
a=rtpmap:101 rex/1000
a=fmtp:101 98/98/98
```

Answer example from a multi-party capable device

```
m=text 12000 RTP/AVP 101 98
a=rtpmap:98 t140/1000
a=rtpmap:101 rex/1000
a=fmtp:101 98/98/98
```

Offer example for both traditional "text/red" and multi-party format:

```
m=text 11000 RTP/AVP 101 100 98
a=rtpmap:98 t140/1000
a=rtpmap:100 red/1000
a=rtpmap:101 rex/1000
a=fmtp:100 98/98/98
a=fmtp:101 98/98/98
```

Answer example from a multi-party capable device

```
m=text 11000 RTP/AVP 101 98
a=rtpmap:98 t140/1000
a=rtpmap:101 rex/1000
a=fmtp:101 98/98/98
```

Offer example for both traditional "text/red" and multi-party format including security:

```
a=fingerprint: SHA-1 \
4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
m=text 11000 RTP/AVP 101 100 98
a=rtpmap:98 t140/1000
a=rtpmap:100 red/1000
a=rtpmap:101 rex/1000
a=fmtp:100 98/98/98
a=fmtp:101 98/98/98
```

The "Fingerprint" is sufficient to offer DTLS-SRTP, with the media line still indicating RTP/AVP.

Answer example from a multi-party capable device including security

```
a=fingerprint: SHA-1 \
FF:FF:FF:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
m=text 11000 RTP/AVP 101 98
a=rtpmap:98 t140/1000
a=rtpmap:101 rex/1000
a=fmtp:101 98/98/98
```

With the "fingerprint" the device acknowledges use of SRTP/DTLS.

Answer example from a multi-party unaware device that also does not support security:

```
m=text 12000 RTP/AVP 100 98
a=rtpmap:98 t140/1000
a=rtpmap:100 red/1000
a=fmtp:100 98/98/98
```

A party who has negotiated the "text/rex" format MUST populate the CSRC-list and format the packets according to the present specification if it acts as an rtp-mixer and sends multi-party text.

A party who has negotiated the "text/rex" capability MUST interpret the contents of the CSRC-list and the packets according to the present specification in received rtp packets using the corresponding payload type.

A party performing as a mixer, which has not negotiated the "text/rex" format, but negotiated a "text/red" or "text/t140" format in a session with a participant SHOULD, if nothing else is specified for the application, format transmitted text to that participant to be suitable to present on a multi-party unaware endpoint as further specified in section [Section 13.2](#).

## 11. Examples

This example shows a symbolic flow of packets from a mixer with loss and recovery. A, B and C are sources of RTT. M is the mixer. Pn indicates primary data in source group "n". Rn1 is first redundant generation data and Rn2 is second redundant generation data in source group "n". A1, B1, A2 etc are text chunks (T140blocks) received from the respective sources. X indicates dropped packet between the mixer and a receiver.



```
|-----|
|Seq no 1      |
|CC=1          |
|CSRC list A   |
|R12: Empty    |
|R11: Empty    |
|P1: A1        |
|-----|
```

Assuming that earlier packets were received in sequence, text A1 is received from packet 1 and assigned to reception area A.

```
|-----|
|Seq no 2      |
|CC=3          |
|CSRC list C,A |
|R12 Empty    |
|R11:Empty    |
|P1: C1       |
|R22 Empty    |
|R21: A1      |
|P2: Empty    |
|-----|
```

Text C1 is received from packet 2 and assigned to reception area C.

```
X-----|
X Seq no 3    |
X CC=2        |
X CSRC list C,A |
X R12: Empty  |
X R11: C1     |
X P1: Empty   |
X R22: A1     |
X R21: Empty  |
X P2: A2      |
X-----|
```

Packet 3 is assumed to be dropped in network problems

```
X-----|
X Seq no 4      |
X CC=3          |
X CSRC list C,B,A|
X R12: Empty    |
X R11: Empty    |
X P1: C2        |
X R22: Empty    |
X R21: Empty    |
X P2: B1        |
X R32: Empty    |
X R31: A2       |
X P3: A3        |
X-----|
```

Packet 4 is assumed to be dropped in network problems

```
X-----|
X Seq no 5      |
X CC=3          |
X CSRC list C,B,A|
X R12: Empty    |
X R11: C2       |
X P1: Empty     |
X R22: Empty    |
X R21: B1       |
X P2: B2       |
X R32: A2       |
X R31: A3       |
X P3: A4       |
X-----|
```

Packet 5 is assumed to be dropped in network problems

```
|-----|
|Seq no 6   |
|CC=3       |
|CSRC list C,B,A |
| R12: C2   |
| R11: Empty |
| P1: Empty  |
| R22: B1    |
| R21: B2    |
| P2: B3     |
| R32: A3    |
| R31: A4    |
| P3: A5     |
|-----|
```

Packet 6 is received. The latest received sequence number was 2. Recovery is therefore tried for 3,4,5. But there is no coverage for seq no 3. A missing text mark (U'FFFD) is created and appended to the common mixer reception area.

For seqno 4, texts C2, B1 and A3 are recovered and appended to their respective reception areas.

For seqno 5, texts B2 and A4 are recovered and appended to their respective reception areas.

Primary text B3 and A5 are received and appended to their respective reception areas.

With only one or two packets lost, there would not be any need to create a missing text marker, and all text would be recovered.

It will be a design decision how to present the missing text markers assigned to the mixer as a source.

## 12. Performance considerations

This specification allows new text from 16 sources per packet. Packets SHOULD be transmitted with a maximum time interval when there is text to be transmitted (either primary or redundant) . The default maximum transmission interval is 300 ms. It is also RECOMMENDED for the mixer to send a packet as soon as text has been received from a source as long as the maximum number of characters per second indicated by the recipient is not exceeded, and also the number of packets sent per second to a recipient is kept under a number. In order to achieve good performance, a receiver for multi-party calls SHOULD declare a sufficient CPS value in SDP for the number of allowable characters per second. These characteristics provide for smooth flow of text with acceptable latency from at least 32 sources simultaneously.

The default maximum rate of reception of real-time text is in RFC 4103 [[RFC4103](#)] specified to be 30 characters per second. The value

MAY be modified in the CPS parameter of the FMT attribute in the media section for RFC 4103. A mixer combining real-time text from a number of sources may have a higher combined flow of text coming from the sources. Endpoints SHOULD therefore specify a suitable higher value for the CPS parameter, corresponding to its real reception capability. A value for CPS of 150 is the default for the "text/rex" format. See RFC 4103 [[RFC4103](#)] for the format and use of the CPS parameter. The same rules apply for the "text/rex" format except for the default value.

### **13. Presentation level considerations**

ITU-T T.140 [[T140](#)] provides the presentation level requirements for the RFC 4103 [[RFC4103](#)] transport. T.140 [[T140](#)] has functions for erasure and other formatting functions and has the following general statement for the presentation:

"The display of text from the members of the conversation should be arranged so that the text from each participant is clearly readable, and its source and the relative timing of entered text is visualized in the display. Mechanisms for looking back in the contents from the current session should be provided. The text should be displayed as soon as it is received."

Strict application of T.140 [[T140](#)] is of essence for the interoperability of real-time text implementations and to fulfill the intention that the session participants have the same information of the text contents of the conversation without necessarily having the exact same layout of the conversation. This also includes the ability to ignore optional presentation control codes not supported by a receiving application.

T.140 [[T140](#)] specifies a set of presentation control codes to include in the stream. Some of them are optional. Implementations MUST be able to ignore optional control codes that they do not support.

There is no strict "message" concept in real-time text. Line Separator SHALL be used as a separator allowing a part of received text to be grouped in presentation. The characters "CRLF" may be used by other implementations as replacement for Line Separator. The "CRLF" combination SHALL be erased by just one erasing action, just as the Line Separator. Presentation functions are allowed to group text for presentation in smaller groups than the line separators imply and present such groups with source indication together with text groups from other sources (see the following presentation examples). Erasure has no specific limit by any delimiter in the text stream.

### **13.1. Presentation by multi-party aware endpoints**

A multi-party aware receiving party, presenting real-time text MUST separate text from different sources and present them in separate presentation areas. The receiving party MAY separate presentation of parts of text from a source in readable groups based on other criteria than line separator and merge these groups in the presentation area when it benefits the user to most easily find and read text from the different participants. The criteria MAY e.g. be a received comma, full stop, or other phrase delimiters, or a long pause.

When text is received from multiple original sources, the presentation SHOULD provide a view where text is added in multiple places simultaneously.

If the presentation presents text from different sources in one common area, the presenting endpoint SHOULD insert text from the local user ended at suitable points merged with received text to indicate the relative timing for when the text groups were completed. In this presentation mode, the receiving endpoint SHALL present the source of the different groups of text.

A view of a three-party RTT call in chat style is shown in this example .

[Alice] Hi, Alice here.		
[Bob] Bob as well.		
[Eve] Hi, this is Eve, calling from Paris.		
I thought you should be here.		
[Alice] I am coming on Thursday, my		
performance is not until Friday morning.		
[Bob] And I on Wednesday evening.		
[Alice] Can we meet on Thursday evening?		
[Eve] Yes, definitely. How about 7pm.		
at the entrance of the restaurant		
Le Lion Blanc?		
[Eve] we can have dinner and then take a walk		
<Eve-typing> But I need to be back to		
the hotel by 11 because I need		
<Bob-typing> I wou		
of course, I underst		

Figure 2: Example of a three-party RTT call presented in chat style.

Other presentation styles than the chat style may be arranged.

This figure shows how a coordinated column view MAY be presented.

Bob	Eve	Alice
		I will arrive by TGV.
My flight is to Orly		Convenient to the main
	Hi all, can we plan	station.
	for the seminar?	
Eve, will you do		
your presentation on		
Friday?	Yes, Friday at 10.	
Fine, wo		We need to meet befo

Figure 3: An example of a coordinated column-view of a three-party session with entries ordered vertically in approximate time-order.

### **13.2. Multi-party mixing for multi-party unaware endpoints**

When the mixer has indicated multi-party capability in an sdp negotiation, but the multi-party capability negotiation fails with an endpoint, then the mixer **SHOULD** compose a best-effort presentation of multi-party real-time text in one stream intended to be presented by an endpoint with no multi-party awareness.

This presentation format has functional limitations and **SHOULD** be used only to enable participation in multi-party calls by legacy deployed endpoints.

The principles and procedures below do not specify any new protocol elements or behaviors. They are instead composed from the information in ITU-T T.140 [T140] and an ambition to provide a best effort presentation on an endpoint which has functions only for two-party calls.

The mixer mixing for multi-party unaware endpoints **SHALL** compose a simulated limited multi-party RTT view suitable for presentation in one presentation area. The mixer **SHALL** group text in suitable groups and prepare for presentation of them by inserting a new line between them if the transmitted text did not already end with a new line. A presentable label **SHOULD** be composed and sent for the source initially in the session and after each source switch. With this procedure the time for source switching is depending on the actions of the users. In order to expedite source switch, a user can for example end its turn with a new line.

#### **13.2.1. Actions by the mixer at reception from the call participants**

When text is received by the mixer from the different participants, the mixer **SHALL** recover text from redundancy if any packets are lost. The mark for lost text [T140ad1] **SHOULD** be inserted in the stream if unrecoverable loss appears. Any Unicode BOM characters, possibly used for keep-alive shall be deleted. The time of arrival of text **SHALL** be stored together with the received text from each source in a queue for transmission to the recipients.

#### **13.2.2. Actions by the mixer for transmission to the recipients**

The following procedure **SHOULD** be applied for each recipient of multi-part text from the mixer.

The text for transmission **SHOULD** be formatted by the mixer for each receiving user for presentation in one single presentation area. Text received from a participant **SHOULD NOT** be included in

transmission to that participant. When there is text available for transmission from the mixer to a receiving party from more than one participant, the mixer SHOULD switch between transmission of text from the different sources at suitable points in the transmitted stream.

When switching source, the mixer SHOULD insert a line separator if the already transmitted text did not end with a new line (line separator or CRLF). A label SHOULD be composed from information in the CNAME and NAME fields in RTCP reports from the participant to have its text transmitted, or from other session information for that user. The label SHOULD be delimited by suitable characters (e.g. '[' ']') and transmitted. The CSRC SHOULD indicate the selected source. Then text from that selected participant SHOULD be transmitted until a new suitable point for switching source is reached.

Seeking a suitable point for switching source SHOULD be done when there is older text waiting for transmission from any party than the age of the last transmitted text. Suitable points for switching are:

- \*A completed phrase ended by comma

- \*A completed sentence

- \*A new line (line separator or CRLF)

- \*A long pause (e.g. > 10 seconds) in received text from the currently transmitted source

- \*If text from one participant has been transmitted with text from other sources waiting for transmission for a long time (e.g. > 1 minute) and none of the suitable points for switching has occurred, a source switch MAY be forced by the mixer at next word delimiter, and also if even a word delimiter does not occur within a time (e.g. 15 seconds) after the scan for word delimiter started.

When switching source, the source which has the oldest text in queue SHOULD be selected to be transmitted. A character display count SHOULD be maintained for the currently transmitted source, starting at zero after the label is transmitted for the currently transmitted source.

There SHOULD be a storage for the latest control code for Select Graphic Rendition (SGR) from each source. If there is an SGR code stored for the current source before the source switch is done, a reset of SGR shall be sent by the sequence SGR 0 [009B 0000 006D] after the new line and before the new label during a source switch. See SGR below for an explanation. This transmission does not



influence the display count. If there is an SGR code stored for the new source after the source switch, that SGR code SHOULD be transmitted to the recipient before the label. This transmission does not influence the display count.

### **13.2.3. Actions on transmission of text**

Text from a source sent to the recipient SHOULD increase the display count by one per transmitted character.

### **13.2.4. Actions on transmission of control codes**

The following control codes specified by T.140 require specific actions. They SHOULD cause specific considerations in the mixer. Note that the codes presented here are expressed in UCS-16, while transmission is made in UTF-8 transform of these codes.

**BEL 0007 Bell** Alert in session, provides for alerting during an active session. The display count SHOULD not be altered.

**NEW LINE 2028** Line separator. Check and perform a source switch if appropriate. Increase display count by 1.

**CR LF 000D 000A** A supported, but not preferred way of requesting a new line. Check and perform a source switch if appropriate. Increase display count by 1.

**INT ESC 0061** Interrupt (used to initiate mode negotiation procedure). The display count SHOULD not be altered.

**SGR 009B Ps 006D** Select graphic rendition. Ps is rendition parameters specified in ISO 6429. The display count SHOULD not be altered. The SGR code SHOULD be stored for the current source.

**SOS 0098** Start of string, used as a general protocol element introducer, followed by a maximum 256 bytes string and the ST. The display count SHOULD not be altered.

**ST 009C** String terminator, end of SOS string. The display count SHOULD not be altered.

**ESC 001B** Escape - used in control strings. The display count SHOULD not be altered for the complete escape code.

**Byte order mark FEFF** Zero width, no break space, used for synchronization and keep-alive. SHOULD be deleted from incoming

streams. Shall be sent first after session establishment to the recipient. The display count shall not be altered.

**Missing text mark FFFD** Replacement character, marks place in stream of possible text loss. SHOULD be inserted by the reception procedure in case of unrecoverable loss of packets. The display count SHOULD be increased by one when sent as for any other character.

**SGR** If a control code for selecting graphic rendition (SGR), other than reset of the graphic rendition (SGR 0) is sent to a recipient, that control code shall also be stored for the source in the storage for SGR. If a reset graphic rendition (SGR 0) originated from a source is sent, then the SGR storage for that source shall be cleared. The display count shall not be increased.

**BS 0008** Back Space, intended to erase the last entered character by a source. Erasure by backspace cannot always be performed as the erasing party intended. If an erasing action erases all text up to the end of the leading label after a source switch, then the mixer must not transmit more backspaces. Instead it is RECOMMENDED that a letter "X" is inserted in the text stream for each backspace as an indication of the intent to erase more. A new line is usually coded by a Line Separator, but the character combination "CRLF" MAY be used instead. Erasure of a new line is in both cases done by just one erasing action (Backspace). If the display count has a positive value it is decreased by one when the BS is sent. If the display count is at zero, it is not altered.

#### **13.2.5. Packet transmission**

A mixer transmitting to a multi-party unaware terminal SHOULD send primary data only from one source per packet. The SSRC SHOULD be the SSRC of the mixer. The CSRC list SHOULD contain one member and be the SSRC of the source of the primary data.

#### **13.2.6. Functional limitations**

When a multi-party unaware endpoint presents a conversation in one display area in a chat style, it inserts source indications for remote text and local user text as they are merged in completed text groups. When an endpoint using this layout receives and presents text mixed for multi-party unaware endpoints, there will be two levels of source indicators for the received text; one generated by the mixer and inserted in a label after each source switch, and another generated by the receiving endpoint and inserted after each

switch between local and remote source in the presentation area. This will waste display space and look inconsistent to the reader.

New text can be presented only from one source at a time. Switch of source to be presented takes place at suitable places in the text, such as end of phrase, end of sentence, line separator and inactivity. Therefore the time to switch to present waiting text from other sources may become long and will vary and depend on the actions of the currently presented source.

Erasure can only be done up to the latest source switch. If a user tries to erase more text, the erasing actions will be presented as letter X after the label.

These facts makes it strongly RECOMMENDED to implement multi-party awareness in RTT endpoints. The use of the mixing method for multi-party-unaware endpoints should be left for use with endpoints which are impossible to upgrade to become multi-party aware.

#### 13.2.7. Example views of presentation on multi-party unaware endpoints

The following pictures are examples of the view on a participant's display for the multi-party-unaware case.

Conference	Alice
	I will arrive by TGV.
[Bob]:My flight is to	Convenient to the main
Orly.	station.
[Eve]:Hi all, can we	
plan for the seminar.	
[Bob]:Eve, will you do	
your presentation on	
Friday?	
[Eve]:Yes, Friday at 10.	
[Bob]: Fine, wo	We need to meet befo

Figure 4: Alice who has a conference-unaware client is receiving the multi-party real-time text in a single-stream. This figure shows how a coordinated column view MAY be presented on Alice's device.

	^
[Alice] Hi, Alice here.	
[mix][Bob] Bob as well.	
[Eve] Hi, this is Eve, calling from Paris	
I thought you should be here.	
[Alice] I am coming on Thursday, my	
performance is not until Friday morning.	
[mix][Bob] And I on Wednesday evening.	
[Eve] we can have dinner and then walk	
[Eve] But I need to be back to	
the hotel by 11 because I need	-
	-
	v
of course, I underst	

Figure 5: An example of a view of the multi-party unaware presentation in chat style. Alice is the local user.

## 14. Gateway Considerations

### 14.1. Gateway considerations with Textphones (e.g. TTYs).

Multi-party RTT sessions may involve gateways of different kinds. Gateways involved in setting up sessions SHALL correctly reflect the multi-party capability or unawareness of the combination of the gateway and the remote endpoint beyond the gateway.

One case that may occur is a gateway to PSTN for communication with textphones (e.g. TTYs). Textphones are limited devices with no multi-party awareness, and it SHOULD therefore be suitable for the gateway to not indicate multi-party awareness for that case. Another solution is that the gateway indicates multi-party capability towards the mixer, and includes the multi-party mixer function for multi-party unaware endpoints itself. This solution makes it possible to make adaptations for the functional limitations of the textphone (TTY).

## **14.2. Gateway considerations with WebRTC.**

Gateway operation to real-time text in WebRTC may also be required. In WebRTC, RTT is specified in draft-ietf-mmusic-t140-usage-data-channel[[I-D.ietf-mmusic-t140-usage-data-channel](#)].

A multi-party bridge may have functionality for communicating by RTT in both RTP streams with RTT and WebRTC t140 data channels. Other configurations may consist of a multi-party bridge with either technology for RTT transport and a separate gateway for conversion of the text communication streams between RTP and t140 data channel.

In WebRTC, it is assumed that for a multi-party session, one t140 data channel is established for each source from a gateway or bridge to each participant. Each participant also has a data channel with two-way connection with the gateway or bridge.

The t140 channel used both ways is for text from the WebRTC user and from the bridge or gateway itself to the WebRTC user. The label parameter of this t140 channel is used as NAME field in RTCP to participants on the RTP side. The other t140 channels are only for text from other participants to the WebRTC user.

When a new participant has entered the session with RTP transport of rtt, a new t140 channel SHOULD be established to WebRTC users with the label parameter composed from the NAME field in RTCP on the RTP side.

When a new participant has entered the multi-party session with RTT transport in a WebRTC t140 data channel, the new participant SHOULD be announced by a notification to RTP users. The label parameter from the WebRTC side SHOULD be used as the NAME RTCP field on the RTP side.

## **15. Updates to RFC 4102 and RFC 4103**

The document updates RFC 4102[[RFC4102](#)] and RFC 4103[[RFC4103](#)] by introducing an extended packet format 'text/rex' for the multi-party mixing case and more strict rules for the use of redundancy, and population of the CSRC list in the packets. Implications for the CSRC list use from RFC 2198[[RFC2198](#)] are hereby not in effect.

## **16. Congestion considerations**

The congestion considerations and recommended actions from RFC 4103 [[RFC4103](#)] are valid also in multi-party situations.

## **17. Acknowledgements**

James Hamlin for format input.

## **18. IANA Considerations**

The IANA is requested to register the media type registration "text/rex" as specified in [Section 9](#). The media type is also requested to be added to the IANA registry for "RTP Payload Format Media Types" <<http://www.iana.org/assignments/rtp-parameters>>.

## **19. Security Considerations**

The RTP-mixer model requires the mixer to be allowed to decrypt, pack and encrypt secured text from the conference participants. Therefore the mixer needs to be trusted. This is similar to the situation for central mixers of audio and video.

The requirement to transfer information about the user in RTCP reports in SDES, CNAME and NAME fields for creation of labels may have privacy concerns as already stated in RFC 3550 [[RFC3550](#)], and may be restricted of privacy reasons. The receiving user will then get a more symbolic label for the source.

## **20. Change history**

### **20.1. Changes included in draft-ietf-avtcore-multi-party-rtt-mix-02**

SDP and better description and visibility of security by OSRTP RFC 8634 needed.

The description of gatewaying to WebRTC extended.

The description of the redundancy header in the packet is improved.

### **20.2. Changes to draft-ietf-avtcore-multi-party-rtt-mix-01**

2,5,6 More efficient format text/rex introduced and attribute a=rtt-mix deleted.

3. Brief about use of OSRTP for security included- More needed.

4. Brief motivation for the solution and why not rtp-translator is used added to intro.

7. More limitations for the multi-party unaware mixing method inserted.

8. Updates to RFC 4102 and 4103 more clearly expressed.

9. Gateway to WebRTC started. More needed.

### **20.3. Changes from draft-hellstrom-avtcore-multi-party-rtt-source-03 to draft-ietf-avtcore-multi-party-rtt-mix-00**

Changed file name to draft-ietf-avtcore-multi-party-rtt-mix-00

Replaced CDATA in IANA registration table with better coding.

Converted to xml2rfc version 3.

### **20.4. Changes from draft-hellstrom-avtcore-multi-party-rtt-source-02 to -03**

Changed company and e-mail of the author.

Changed title to "RTP-mixer formatting of multi-party Real-time text" to better match contents.

Check and modification where needed of use of RFC 2119 words SHALL etc.

More about the CC value in sections on transmitters and receivers so that 1-to-1 sessions do not use the mixer format.

Enhanced section on presentation for multi-party-unaware endpoints

A paragraph recommending CPS=150 inserted in the performance section.

### **20.5. Changes from draft-hellstrom-avtcore-multi-party-rtt-source-01 to -02**

In Abstract and 1. Introduction: Introduced wording about regulatory requirements.

In section 5: The transmission interval is decreased to 100 ms when there is text from more than one source to transmit.

In section 11 about sdp negotiation, a SHOULD-requirement is introduced that the mixer should make a mix for multi-party unaware endpoints if the negotiation is not successful. And a reference to a later chapter about it.

The presentation considerations chapter 14 is extended with more information about presentation on multi-party aware endpoints, and a new section on the multi-party unaware mixing with low functionality but SHOULD a be implemented in mixers. Presentation examples are added.

A short chapter 15 on gateway considerations is introduced.

Clarification about the text/t140 format included in chapter 10.

This sentence added to the chapter 10 about use without redundancy.  
"The text/red format SHOULD be used unless some other protection against packet loss is utilized, for example a reliable network or transport."

Note about deviation from RFC 2198 added in chapter 4.

In chapter 9. "Use with SIP centralized conferencing framework" the following note is inserted: Note: The CSRC-list in an RTP packet only includes participants who's text is included in one or more text blocks. It is not the same as the list of participants in a conference. With audio and video media, the CSRC-list would often contain all participants who are not muted whereas text participants that don't type are completely silent and so don't show up in RTP packet CSRC-lists.

## **20.6. Changes from draft-hellstrom-avtcore-multi-party-rtt-source-00 to -01**

Editorial cleanup.

Changed capability indication from fmtp-parameter to sdp attribute "rtt-mix".

Swapped order of redundancy elements in the example to match reality.

Increased the SDP negotiation section

## **21. References**

### **21.1. Normative References**

#### **[I-D.ietf-mmusic-t140-usage-data-channel]**

Holmberg, C. and G. Hellstrom, "T.140 Real-time Text Conversation over WebRTC Data Channels", Work in Progress, Internet-Draft, draft-ietf-mmusic-t140-usage-data-channel-14, 10 April 2020, <<https://tools.ietf.org/html/draft-ietf-mmusic-t140-usage-data-channel-14>>.

**[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>>.

- [RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J.C., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, DOI 10.17487/RFC2198, September 1997, <<https://www.rfc-editor.org/info/rfc2198>>.
- [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>>.
- [RFC4102] Jones, P., "Registration of the text/red MIME Sub-Type", RFC 4102, DOI 10.17487/RFC4102, June 2005, <<https://www.rfc-editor.org/info/rfc4102>>.
- [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>>.
- [RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <<https://www.rfc-editor.org/info/rfc4855>>.
- [RFC6263] Marjou, X. and A. Sollaud, "Application Mechanism for Keeping Alive the NAT Mappings Associated with RTP / RTP Control Protocol (RTCP) Flows", RFC 6263, DOI 10.17487/RFC6263, June 2011, <<https://www.rfc-editor.org/info/rfc6263>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [RFC8643] Johnston, A., Aboba, B., Hutton, A., Jesske, R., and T. Stach, "An Opportunistic Approach for Secure Real-time Transport Protocol (OSRTP)", RFC 8643, DOI 10.17487/RFC8643, August 2019, <<https://www.rfc-editor.org/info/rfc8643>>.
- [T140] ITU-T, "Recommendation ITU-T T.140 (02/1998), Protocol for multimedia application text conversation", February 1998, <<https://www.itu.int/rec/T-REC-T.140-199802-I/en>>.
- [T140ad1] ITU-T, "Recommendation ITU-T.140 Addendum 1 - (02/2000), Protocol for multimedia application text conversation",

February 2000, <<https://www.itu.int/rec/T-REC-T.140-200002-I!Add1/en>>.

## 21.2. Informative References

- [RFC4353] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", RFC 4353, DOI 10.17487/RFC4353, February 2006, <<https://www.rfc-editor.org/info/rfc4353>>.
- [RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, Ed., "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, DOI 10.17487/RFC4575, August 2006, <<https://www.rfc-editor.org/info/rfc4575>>.
- [RFC4579] Johnston, A. and O. Levin, "Session Initiation Protocol (SIP) Call Control - Conferencing for User Agents", BCP 119, RFC 4579, DOI 10.17487/RFC4579, August 2006, <<https://www.rfc-editor.org/info/rfc4579>>.

## Author's Address

Gunnar Hellstrom  
Gunnar Hellstrom Accessible Communication  
Esplanaden 30  
SE-13670 Vendelso  
Sweden

Email: [gunnar.hellstrom@ghaccess.se](mailto:gunnar.hellstrom@ghaccess.se)