## Source: Huawei Technologies Co., Ltd (Rapporteur)

## Title: [MP\_RTT] Proposed Permanent Document

## Version: 0.3.0

## Agenda Item: 14.11

## Document for: Discussion and Agreement

Table of Contents

[1. Introduction 2](#_Toc132964874)

[2. Introduction to Multiparty RTT 2](#_Toc132964875)

[2.1 Multiparty RTT use cases and scenarios 2](#_Toc132964876)

[2.2 Multiparty RTT requirements 3](#_Toc132964877)

[3. Multiparty RTT Solutions 4](#_Toc132964878)

[3.1 Multiparty RTT over RTP Solution 4](#_Toc132964879)

[3.2 Multiparty RTT over IMS Data Channel Solution 4](#_Toc132964880)

[4. Comparison between RTP and IMS Data Channel Solution 15](#_Toc132964881)

[5. Interworking for Multiparty RTT between RTP and IMS Data Channel Solution 15](#_Toc132964882)

[6. KPIs 15](#_Toc132964883)

[7. References 16](#_Toc132964884)

# Introduction

The use cases and requirements proposed in [SA4R230052](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/3GPP_SA4_AHOC_MTGs/SA4_RTC/Docs/S4aR230052.zip) were agreed during RTC SWG post 122 #9 telco and the proposed permanent document update in [S4aR230061](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/3GPP_SA4_AHOC_MTGs/SA4_RTC/Docs/S4aR230061.zip) was agreed upon during RTC SWG post 122#10 telco. In SA4#123-e, [S4-230704](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_123-e/Docs/S4-230704.zip) was also agreed and its contents were integrated in S4-230702.

This version (v0.2.1) implemented some minor editorial changes.

The objective of this work item is to specify suitable solutions for multiparty RTT media in IMS, for both RTP and IMS data channel transport.

The concrete objectives are as follows:

* Collect and document detailed use cases for multiparty usage of RTT
* Develop harmonized solutions for both RTP and IMS data channel transport that address the detailed use case needs
* Amend existing IMS control/signalling flows to support the solutions, if found necessary
* Document pros and cons of each solution, and provide implementation guidelines to equipment vendors, as an informational Annex
* Inform/coordinate with at least SA2, CT1, CT4, and with other relevant 3GPP groups as found necessary, to enable alignment and possible updates of specifications under the responsibility of those groups.

# Introduction to Multiparty RTT

### 2.1 Multiparty RTT use cases and scenarios

According to clause 6.4 of [Draft - DTR/HF-00103708 v0.0.11](https://docbox.etsi.org/HF/HF/05-CONTRIBUTIONS/2022/HF(22)088017_Draft_-_DTR_HF-00103708_v0_0_11_TR_103_708.zip) [1], Multiparty RTT use cases

are defined as follows:

* Call using RTT within a small group of Deaf persons
* Deaf person calling emergency service and using RTT
* Hard-of-hearing user talking with hearing friends
* Deaf user participating in conference getting transcription support
* Deaf user participating in conference contributing by text-to-speech
* Deaf-Blind user participating in remote meeting
* Person in a critical situation making an emergency call by RTT
* Person in remote group meeting in occasional noise
* Relay service using multiparty technology
* Using an RTT relay service to connect to a voice conference call

### 2.2 Multiparty RTT requirements

2.2.1 General requirements

The general Multiparty RTT requirements from existing standards are listed as follows:

* A solution shall be applicable to IMS as specified in 3GPP TS 23.228 [3], Additionally, 3GPP TS 24.147 [5] provides the protocol details for conferencing within IMS based on SIP, SIP Events, SDP and the Binary Floor Control BFCP.
* If text loss is detected or suspected, a missing text marker should be inserted in the text stream as defined in ITU-T T.140 Amendment 1 [6].
* The display of text from the members of the conversation shall 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 as defined in ITU-T T.140 [6].
* It MUST be possible to use real-time text in conferences both as a medium of discussion between individual participants (for example, for sidebar discussions in real-time text while listening to the main conference audio) and for central support of the conference with real-time text interpretation of speech. Further session setup and control requirements can be found in RFC5194 [7].

2.2.2 Performance requirements

The Multiparty RTT performance requirements from existing standards are listed as follows:

* The mixer performance requirements can be expressed in one number, extracted from the user requirements on real-time text expressed in ITU-T F.700 [8], where it is stated that for "good" usability, text characters should not be delayed more than 1 second from creation to presentation. For "usable" usability the figure is 2 seconds.
* If buffering is provided in the data channel, it should not delay transmission more than 500 ms [3]. A buffering time of 300 ms is RECOMMENDED when the application or end-to-end network conditions are not known to require another value as indicated in RFC 4103 [9].

# Multiparty RTT Solutions

### 3.1 Multiparty RTT over RTP Solution

#### 3.1.1 Architecture



Figure *2*.1.1-1 Multi-party RTT over RTP architecture

The Multi-party RTT over RTP solution can reuse the current architecture, which is defined in clause 4 of TS 23.228[1].

According to clause 1.2 of RFC9071[2], for multiparty considerations, several alternatives were introduced, but only two alternatives were selected when searching for an efficient and easily implemented multiparty method for real-time text:

RTP-mixer-based method for multiparty-aware endpoints:

This solution is used when the endpoint supports multiparty-aware identifying by “a=rtt-mixer” in the SDP negotiation procedure. Only one single RTP stream for each participant, the source is indicated in the CSRC element in the RTP packets. Text from one source shall be transmitted in the same packet if available for transmission at the same time. Text from different sources must not be transmitted in the same packet.

Pros:

Good performance for multiparty RTT communication with real time transmission.

Cons:

Has new requirements on the endpoint.

Mixing for multiparty-unaware endpoints:

This solution is used as a fallback solution when the receiving endpoint is not capable of handling the mixed format. This is made possible by having the mixer insert a new line and a text-formatted source label before each switch of text source in the stream. Switching the source can only be done in places in the text where it does not disturb the perception of the contents. Text from only one source at a time can be presented in real time. The delay will therefore vary.

Pros:

No need modifications in existing user devices implementing RFC4103[3] for real-time text.

Cons:

Text from only one source at a time can be presented in real time. The delay will therefore vary.

### 3.1.2 Call Flow

#### 3.1.2.1 SDP Negotiation for RTT-mixed-based multiparty Procedure



Figure 2.1.2.1-1 RTT-mixed SDP negotiation between two parties

The main steps are shown as below:

1. If the Caller party supports RTP-mixer-based method, when the caller party initiates an SDP offer, it can add “a=rtt-mixer” in “m=text” line. The SDP example is shown as below:

m=text 11000 RTP/AVP 100 98

a=rtpmap:98 t140/1000

a=fmtp:98 cps=90

a=rtpmap:100 red/1000

a=fmtp:100 98/98/98

a=rtt-mixer

2-3. If the called party supports RTP-mixer-based method, when the called party receives an SDP offer containing “a=rtt-mixer” in “m=text” line, it should include “a=rtt-mixer” in the corresponding “m=text” line in the SDP answer. The SDP example is shown as below:

m=text 14000 RTP/AVP 100 98

a=rtpmap:98 t140/1000

a=fmtp:98 cps=90

a=rtpmap:100 red/1000

a=fmtp:100 98/98/98

a=rtt-mixer

4-5. If the called party doesn’t support RTP-mixer-based method, when the called party receives an SDP offer containing “a=rtt-mixer” in “m=text” line, it should remove “a=rtt-mixer” in the corresponding “m=text” line in the SDP answer. The SDP example is shown as below:

m=text 14000 RTP/AVP 100 98

a=rtpmap:98 t140/1000

a=fmtp:98 cps=90

a=rtpmap:100 red/1000

a=fmtp:100 98/98/98



Figure 2.1.2.1-2 RTT-mixed SDP negotiation for Multiparty

The main steps are shown as below:

1-2. UE-A creates a conference with UE-B and UE-C.

3. UE-A will finish SDP negotiation with MRF, the RTT-mixed SDP negotiation procedure is the same as Figure 2.1.2.1-1.

4-6. UE-A invites UE-B to the conference, UE-A sends a REFER message to IMS, IMS will finish SDP negotiation with UE-B, the RTT-mixed SDP negotiation procedure is the same as Figure 2.1.2.1-1.

7-9. UE-A invites UE-C to the conference, UE-A sends a REFER message to IMS, IMS will finish SDP negotiation with UE-C, the RTT-mixed SDP negotiation procedure is the same as Figure 2.1.2.1-1.

10-11. UE-D joins the conference, IMS will finish SDP negotiation with UE-D, the RTT-mixed SDP negotiation procedure is the same as Figure 2.1.2.1-1.

#### 3.1.2.2 Multiparty RTT Processing Procedure



The main steps are shown as below:

UE-B support RTT-mixer-based method, but UE-C can’t support. UE-A, UE-B and UE-C enter a multi-party RTT conference.

1. UE-A sends RTT in the conference, the RTT content in RTP packet should follow RFC4103[3].

2. MRF acts as a mixer, and MRF will decide how to handle the RTT content based on the SDP negotiation on rtt-mixer with UE-B and UE-C.

3. For UE-B that supports RTT-mixer-based method, MRF will modify the RTP packets, set CC=1, and put UE-A in the CSRC list. An example is shown as below:

|Seq no 101, Time=20400 |

|CC=1 |

|CSRC list A |

|R2: Empty, Offset=600 |

|R1: Empty, Offset=300 |

|P: A1 |

4. For UE-C that does not support RTT-mixer-based method, MRF will treat it as multiparty-unaware endpoint, a presentable label be composed and sent for the source initially in the session and after each source switch. An example is shown as below:

|Seq no 101, Time=20400 |

|CC=0 |

|SSRC |

|R2: Empty, Offset=600 |

|R1: Empty, Offset=300 |

|P: [UE-A]A1 |

### 3.2 Multiparty RTT over IMS Data Channel Solution

#### 3.2.1 Architecture



Figure 3.2.1-1 Multiparty RTT over IMS Data Channel Architecture

The Multiparty RTT over data channel solution is based on data channel architecture, which is defined in clause AC.2.1 of TS 23.228 [3].

According to clause 5.5 of RFC8865 [4], for multiparty considerations, two alternatives were considered when searching for an efficient and easily implemented multiparty method for real-time text:

Multiple DC streams, one per participant:

One DC stream per source would be sent in the same session. UE can identify the source by the “label” attribute in the DC stream ID line when receiving RTT. If a new UE is added to the conference, a new downlink stream ID indicating the new UE should be added to all the existing participants. The conference application needs to manage the mapping relationship between the UE identity and the steam ID of each participant, obtain the corresponding UE identity according to the stream ID when receiving the real-time text of each participant.

**Pros:**

This is a straightforward solution. The load per source is low.

**Cons:**

With a high number of participants, the overhead of establishing and maintaining the high number of data channels required may be high, even if the load per channel is low.

Single DC stream, each participate use only one DC stream:

Only one DC stream for each participate, no SIP negotiation procedure for each participant when a new UE is added to the conference. The conference server should add a source information in front of the RTT content by identifying the label attribute in the DC stream ID line when receiving RTT from a UE.

**Pros:**

No negotiation when a new UE is added to the conference.

**Cons:**

The conference server should add decode and re-encode the RTT content.

#### 3.2.2 Call Flow

##### 3.2.2.1 Multi DC Streams

An example for three participants in a conference:



Figure 3.2.2.1-1 Multi DC Streams Example

Each UE has one uplink stream ID and two downlink stream IDs, if a new UE is added to the conference, a new downlink stream ID indicating the new UE should be added to all the existing participants.

The conference application needs to manage the mapping relationship between the UE name and the steam ID of each participant, obtain the corresponding UE identity according to the stream ID when receiving the real-time text of each participant, and add the UE identity before the real-time text to correctly display the source.

##### 3.2.2.1.1 UE Aware Mode



Figure 3.2.2.1.1-1 Multi DC Streams with UE Aware Mode Call Flow

The steps are shown as below:

Case 1: UE-A create a conference and join UE-B and UE-C into the conference, then run the RTT application.

1. UE-A, UE-B and UE-C enter an audio/video conference and download the RTT application on each participant.

2. The UE-A runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including 3 DC stream IDs, one ‘sendonly’ for UE-A sending RTT to other participants, one ‘recvonly’ for receiving UE-B’s RTT, and the last one ‘recvonly’ for receiving UE-C’s RTT, the label attribute in each ‘a=dcmap’ can be get from the conference information, which can identify each DC stream belongs to whom. The SDP offer example is shown as below:

m=application 911 UDP/DTLS/SCTP webrtc-datachannel

c=IN IP6 2001:db8::3

a=max-message-size:1000

a=sctp-port 5000

a=setup:actpass

a=dcmap:200 label="A-Identity";subprotocol="t140"

a=dcsa:200 fmtp:t140 cps=20 sendonly

a=dcsa:200 hlang-send:es eo

a=dcmap:201 label="B-Identity";subprotocol="t140"

a=dcsa:201 fmtp:t140 cps=20 recvonly

a=dcsa:201 hlang-recv:es eo

a=dcmap:202 label="C-Identity";subprotocol="t140"

a=dcsa:202 fmtp:t140 cps=20 recvonly

a=dcsa:202 hlang-recv:es eo

3. DCSF establishes corresponding DC stream IDs for UE-A.

4-5. IMS-A sends an REINVITE message with three stream IDs to UE-B and establish corresponding DC stream IDs for UE-B. The stream IDs are similar to step2.

6-7. IMS-A sends an REINVITE message with three stream IDs to UE-C and establish corresponding DC stream IDs for UE-C. The stream IDs are similar to step2.

Case 2: UE-D call into the conference and run the RTT application.

8. UE-D calls into the conference created by UE-A, and runs the RTT application.

9. UE-D runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one DC stream ID for UE-D sending RTT to other participants.

10. IMS-A establishes the DC stream for UE-D.

11. The IMS-A identifies that there are three participants in the conference, so IMS-A decides to add a new downlink DC stream for each participant, and finally add three downlink streams for UE-D.

12-14. The IMS-A adds a new downlink DC stream for UE-A/UE-B/UE-C simultaneously, for receiving UE-D’s RTT.

15. The IMS-A adds three downlink DC streams for UE-D, for receiving UE-A/UE-B/UE-C’s RTT.

When UE-A sends RTT over the uplink stream ID, DCMF/MRF will simultaneously send the RTT to UE-B, UE-C and UE-D through the dedicated stream ID channel, UE-B, UE-C and UE-D can identify the source by the corresponding “label” attribute that included in the ‘a=dcmap’ line.

##### 3.2.2.1.2 UE Unaware Mode



Figure 3.2.2.1.2-1 Multi DC Streams with UE Unaware Mode Call Flow

The steps are shown as below:

Case 1: UE-A creates a conference and joins UE-B and UE-C into the conference, then runs the RTT application.

1. UE-A, UE-B and UE-C enter an audio/video conference and download the RTT application on each participant.

2. The UE-A runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one uplink DC stream ID with ‘sendonly’ for UE-A sending RTT to other participants, the label attribute in ‘a=dcmap’ can be get from UE-A’s identity. The SDP offer example is shown as below:

m=application 911 UDP/DTLS/SCTP webrtc-datachannel

c=IN IP6 2001:db8::3

a=max-message-size:1000

a=sctp-port 5000

a=setup:actpass

a=dcmap:200 label="A-Identity";subprotocol="t140"

a=dcsa:200 fmtp:t140 cps=20 sendonly

a=dcsa:200 hlang-send:es eo

3. DCSF establishes corresponding DC stream ID for UE-A.

4. The IMS-A identifies that there are three participants in the conference, so IMS-A decides to add another two new downlink DC streams for UE-A, and three DC streams including one uplink DC streams and two downlink DC streams for the other participants.

5-6. IMS-A sends an REINVITE message adding two downlink stream IDs to UE-A and establish corresponding DC stream IDs for UE-A. The SDP offer example is shown as below:

m=application 911 UDP/DTLS/SCTP webrtc-datachannel

c=IN IP6 2001:db8::3

a=max-message-size:1000

a=sctp-port 5000

a=setup:actpass

a=dcmap:200 label="A-Identity";subprotocol="t140"

a=dcsa:200 fmtp:t140 cps=20 recvonly

a=dcsa:200 hlang-send:es eo

a=dcmap:201 label="B-Identity";subprotocol="t140"

a=dcsa:201 fmtp:t140 cps=20 sendonly

a=dcsa:201 hlang-recv:es eo

a=dcmap:202 label="C-Identity";subprotocol="t140"

a=dcsa:202 fmtp:t140 cps=20 sendonly

a=dcsa:202 hlang-recv:es eo

7-8. IMS-A sends an REINVITE message with three stream IDs to UE-B and establish corresponding DC stream IDs for UE-B. The stream IDs are similar to step4.

9-10. IMS-A sends an REINVITE message with three stream IDs to UE-C and establish corresponding DC stream IDs for UE-C. The stream IDs are similar to step4.

Case 2: UE-D calls into the conference and run the RTT application.

11. UE-D calls into the conference created by UE-A, and runs the RTT application.

12. UE-D runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one DC stream ID for UE-D sending RTT to other participants.

13. IMS-A establishes the DC stream for UE-D.

14. The IMS-A identifies that there are three participants in the conference, so IMS-A decides to add a new downlink DC stream for each participant, and finally add three downlink streams for UE-D.

15-17. The IMS-A adds a new downlink DC stream for UE-A/UE-B/UE-C simultaneously, for receiving UE-D’s RTT.

18. The IMS-A adds three downlink DC streams for UE-D, for receiving UE-A/UE-B/UE-C’s RTT.

When UE-A sends RTT over the uplink stream ID, DCMF/MRF will simultaneously send the RTT to UE-B, UE-C and UE-D through the dedicated stream ID channel, UE-B, UE-C and UE-D can identify the source by the corresponding “label” attribute that included in the ‘a=dcmap’ line.

#### 3.2.2.2 Single DC Stream

An example for three participants in a conference:



Figure 3.2.2.2-1 Single DC Stream Example

T140 protocol is too old to be extended to support adding the source label, so the conference server can add a source label getting from the “label” attribute of ‘a=dcmap’ line in front of the text content when receiving the real-time text from a UE, and the terminal can display it directly without modification.



Figure 3.2.2.2-2 Single DC Stream Call Flow

The steps are shown as below:

Case 1: UE-A create a conference and join UE-B and UE-C into the conference, then run the RTT application.

1. UE-A, UE-B and UE-C enter an audio/video conference and download the RTT application on each participant.

2. The UE-A runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including only one DC stream ID, the SDP offer example:

m=application 911 UDP/DTLS/SCTP webrtc-datachannel

c=IN IP6 2001:db8::3

a=max-message-size:1000

a=sctp-port 5000

a=setup:actpass

a=dcmap:200 label="A-Identity";subprotocol="t140"

a=dcsa:200 fmtp:t140 cps=20 sendrecv

a=dcsa:200 hlang-send:es eo

3. DCSF establishes corresponding DC stream ID for UE-A.

4-5. IMS-A sends an REINVITE message with only one stream ID to UE-B and establishes corresponding DC stream ID for UE-B. The stream ID is similar to step2.

6-7. IMS-A sends an REINVITE message with only one stream ID to UE-C and establishes corresponding DC stream ID for UE-C. The stream ID is similar to step2.

Case 2: UE-D call into the conference and run the RTT application.

8. UE-D calls into the conference created by UE-A, and runs the RTT application.

9. UE-D runs application and sends an REINVITE message to establish application data channel, this REINVITE message carries SDP including one DC stream ID for UE-D sending and receiving RTT.

10. IMS-A establishes the DC stream for UE-D.

When UE-A sends RTT over the uplink stream ID, DCMF/MRF will identify the source by the application data channel established is between UE-A and DCMF/MRF, and then add the UE-A’s identity as source to the RTT content. DCMF simultaneously send the RTT to UE-B, UE-C and UE-D through the dedicated stream ID channel, UE-B, UE-C and UE-D directly display the RTT content.

# Comparison between RTP and IMS Data Channel Solution

# Interworking for Multiparty RTT between RTP and IMS Data Channel Solution

# KPIs

# References

------------------------------------------------------------------------------

[1] Draft - DTR/HF-00103708 v0.0.11: “Human Factors (HF);Real-Time Text (RTT) in Multiparty conference calling”

[2] draft-hellstrom-avtcore-Multiparty-rtt-solutions-08: “Real-time text solutions for multiparty sessions”

[3] TS 23.228: “IP Multimedia Subsystem (IMS)”

[4] RFC8865: “T.140 Real-Time Text Conversation over WebRTC Data Channels”

[5] 3GPP TS 24.147 Conferencing using the IP Multimedia (IM) Core Network (CN) subsystem; Stage 3 (Release 17), 03/2022

[6] ITU-T T.140 Protocol for multimedia application text conversation, 02/1998, and T.140 Addendum 1, 02/2000

[7] IETF RFC 5194 Framework for Real-Time Text over IP Using the Session Initiation Protocol (SIP), 06/2008

[8] ITU-T F.700 Framework Recommendation for multimedia services, 11/2000

[9] IETF RFC 4103 RTP Payload for Text Conversation, 05/2005

[10] ETSI TS 103 479 Emergency Communications (EMTEL); Core elements for network independent access to emergency services, V1.1.1, 12/2019

[11] IETF RFC 9071 RTP-Mixer Formatting of Multiparty Real-Time Text, 07/2021

[12] ETSI TS 101 470 Emergency Communications (EMTEL); Total Conversation Access to Emergency Services, V1.0.3, 06/2021

[13] ETSI TR 103 201 Emergency Communications (EMTEL); Total Conversation for emergency communications; implementation guidelines, V1.1.1, 03/2016