ETSI TR 126 982 V18.0.0 (2024-05)

TECHNICAL REPORT

5G;
Implementation guidelines for Multiparty RTT
(3GPP TR 26.982 version 18.0.0 Release 18)

Reference

RTR/TSGS-0426982vi00

Keywords

5G

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under https://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

shall indicates a mandatory requirement to do something

shall not indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

should indicates a recommendation to do something

should not indicates a recommendation not to do something

may indicates permission to do something

need not indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

can indicates that something is possible

cannot indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

will indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

will not indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

might indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

might not indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

is (or any other verb in the indicative mood) indicates a statement of fact

is not (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

1 Scope

The present document provides the protocol details for the Multiparty Real-Time Text (Multiparty RTT) service in the IP Multimedia (IM) Core Network (CN) subsystem based on the requirements from 3GPP TS 22.173 [2].

The Multiparty RTT service is an operator specific service by which an operator enables the subscribers in conference to use real-time text during the conference session.

The present document is applicable to User Equipment (UE) and Application Servers (AS) which are intended to support the Multiparty RTT service.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 22.173: "Multimedia Telephony Service and supplementary services".

[3] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[4] 3GPP TS 24.147: "Conferencing Using IP Multimedia Core Network; Stage 3".

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

[6] IETF RFC 5194(2008) “Framework for Real-Time Text over IP Using the Session Initiation Protocol (SIP)”

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

[8] IETF RFC 9071(2021): “RTP-Mixer Formatting of Multiparty Real-Time Text”

[9] IETF RFC 8865 (2021): “T.140 Real-Time Text Conversation over WebRTC Data Channels”

[10] IETF RFC 4103 (2005): "RTP Payload for Text Conversation", G. Hellstrom and P. Jones.

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

3.2 Symbols

void

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

AS	Application Server
BFCP	Binary Floor Control Protocol
CSCF	Call Session Control Function
DC	Data Channel
IMS	IP Multimedia Subsystem
ITU-T	International Telecommunication Union-Telecommunication Standardization Sector
MRF	Multimedia Resource Function
MTSI	Multimedia Telephony Service over IMS
PRACK	Provisional Response Acknowledgement
RTC	Real-Time Communication
RTP	Real-time Transport Protocol
RTT	Real-Time Text
SDP	Session Description Protocol
SIP	Session Initiation Protocol
UE	User Equipment
WebRTC	Web Real-Time Communication

4 Introduction of multiparty real-time text (Multiparty RTT)

4.1 General

Real-time text (RTT) may be used during conference sessions so that all call participants are enabled to create and send RTT to the other participants, and the text being presented in readable chunks growing in real time with indication of source. The presentation provides an approximate view of the relative timing of text from different parties.

The MTSI client in terminal may support multiparty real-time text (Multiparty RTT) as defined in this clause, the Multiparty RTT functionality has two solutions, over RTP and over IMS data channel. For the case of DCMTSI client in terminal, only the IMS data channel based solution is applicable.

The Multiparty RTT functionality for MTSI enables support of real-time text communication between each participant in a conference session. It addresses scenarios in some special groups of people, e.g., deaf person calls emergency service using RTT.

4.2 Study on Multiparty RTT requirements from other standards

4.2.1 General requirements

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

- A solution should be applicable to IMS as specified in 3GPP TS 23.228 [3]. Additionally, 3GPP TS 24.147 [4] 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[5].

- 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 as defined in ITU-T T.140 [5].

- It should 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 [6].

4.2.2 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 [7], 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 500ms. A buffering time of 300ms is recommended when the application or end-to-end network conditions are not known to require another value as indicated in RFC 4103[10].

5 Possible solutions to enable Multiparty RTT over RTP

5.1 Architecture considerations

The Multiparty RTT over RTP solution can reuse the existing architecture, which is defined in clause 4 of TS 23.228[3].

According to clause 1.2 of RFC 9071[8], 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:

1) 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 should be transmitted in the same packet if available for transmission at the same time. Text from different sources should not be transmitted in the same packet.

   The main advantage of this solution is that it provides good performance for multiparty RTT communication with real time transmission. But it also creates new requirements on the endpoint. For the endpoint that support multiparty-aware, this solution should be implemented.

2) 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.

   The main advantage of this solution is no need modifications in existing user devices implementing RFC4103[10] for real-time text. But the text from only one source at a time can be presented in real time. The delay will therefore vary.

5.2 Possible procedures

5.2.1 RTT-mixed SDP negotiation between two parties Procedure

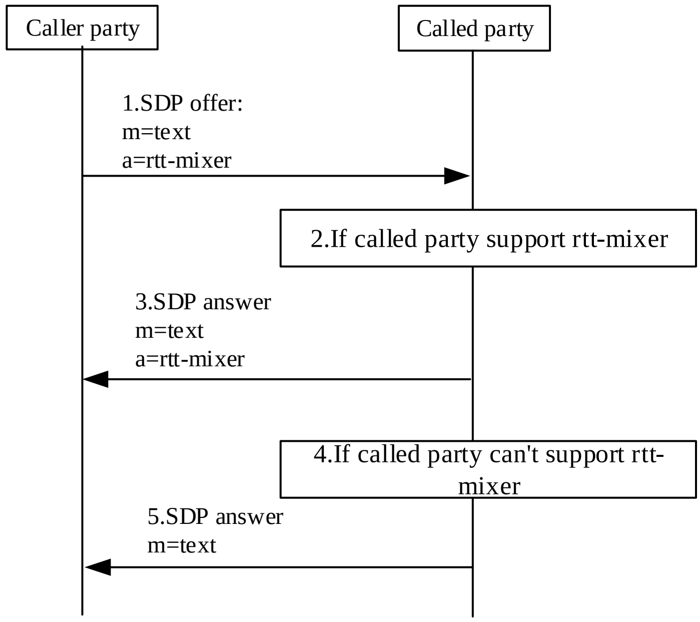

Figure 5.2.1-1 RTT-mixed SDP negotiation between two parties

The main steps for RTT-mixed SDP negotiation between two parties 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:

Table 5.2.1.1: SDP example

SDP offer
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 would include “a=rtt-mixer” in the corresponding “m=text” line in the SDP answer. The SDP example is shown as below:

Table 5.2.1.2: SDP example

SDP answer
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 would remove “a=rtt-mixer” in the corresponding “m=text” line in the SDP answer. The SDP example is shown as below:

Table 5.2.1.3: SDP example

SDP answer
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
```

5.2.2 RTT-mixed SDP negotiation for Multiparty Procedure

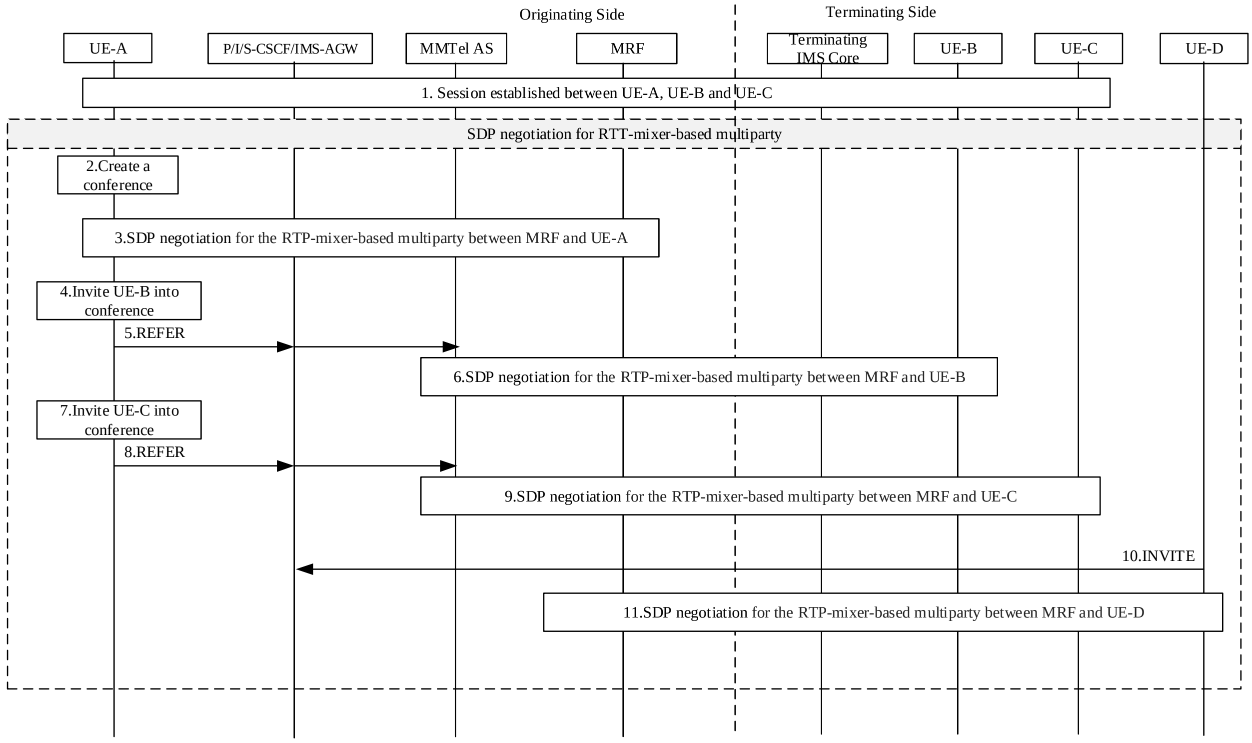

Figure 5.2.2-1 RTT-mixed SDP negotiation for Multiparty

The main steps for RTT-mixed SDP negotiation for Multiparty are shown as below:

1-2. UE-A creates a conference.

3. UE-A will finish SDP negotiation with MRF, the RTT-mixed SDP negotiation procedure is the same as Figure 5.2.1-1.

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 5.2.1-1.

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 5.2.1-1.

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 5.2.1-1.

5.2.3 Multiparty RTT Processing Procedure

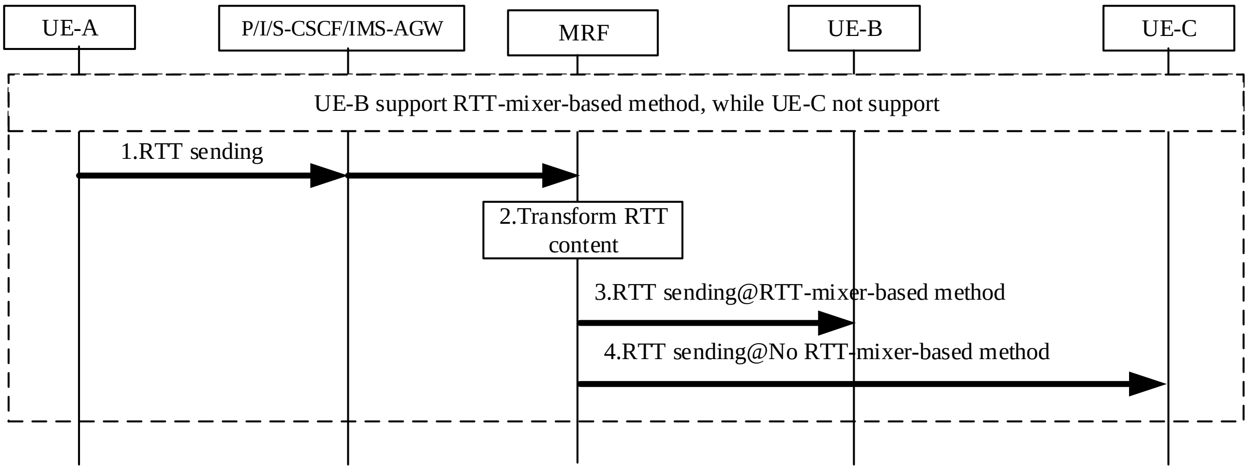

Figure 5.2.3-1 Multiparty RTT over RTP Processing

Taking into consideration that some UEs might not support RTT, e.g. UE-A and UE-B support RTT-mixer-based method, but UE-C cannot support, while UE-A, UE-B and UE-C enter same multi-party RTT conference.

As illustrated in figure 5.2.3-1,the main steps for these typical scenarios are shown as below:

1. UE-A sends RTT in the conference, the RTT content in RTP packet would follow RFC 4103[10].

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 |
```

6 Possible solutions to enable Multiparty RTT over IMS data channel

6.1 Architecture considerations

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 [9], for multiparty considerations, two alternatives were considered when searching for an efficient and easily implemented multiparty method for real-time text, Only one can be implemented without further standardization and is therefore specified here.

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 would 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.

The main advantage of this solution is, as a straightforward solution, the load per source is low. But 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.

6.2 Possible procedures

6.2.1 Multi DC Streams

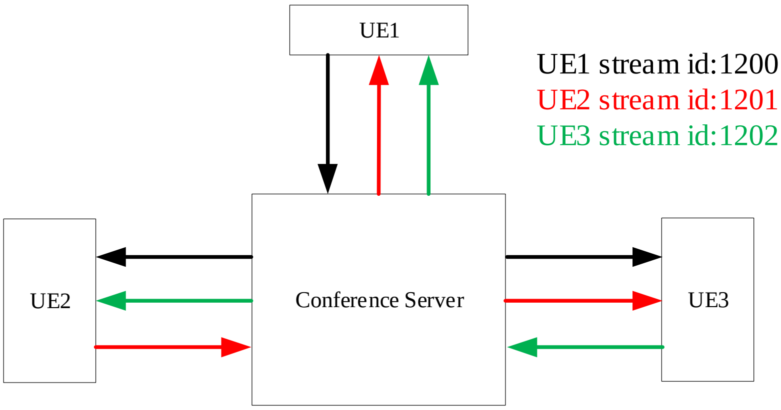

Figure 6.2.1-1 Multi DC Streams Example

An example for three participants in a conference as shown in figure 6.2.1-1, 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 would be added to all the existing participants.

The conference server (MF/MRF or DC AS) 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, and add the UE identity before the real-time text to correctly display the source.

The call flows for different UE modes are shown below separately:

1) UE Aware Mode: The conference creator needs to be aware of who are in the conference, and carry each participant's user name when initiates ADC establishment.

2) UE Unaware Mode: The conference creator needs to be aware of who are in the conference, only carry itself user name when initiates ADC establishment.

6.2.1.1 UE Aware Mode

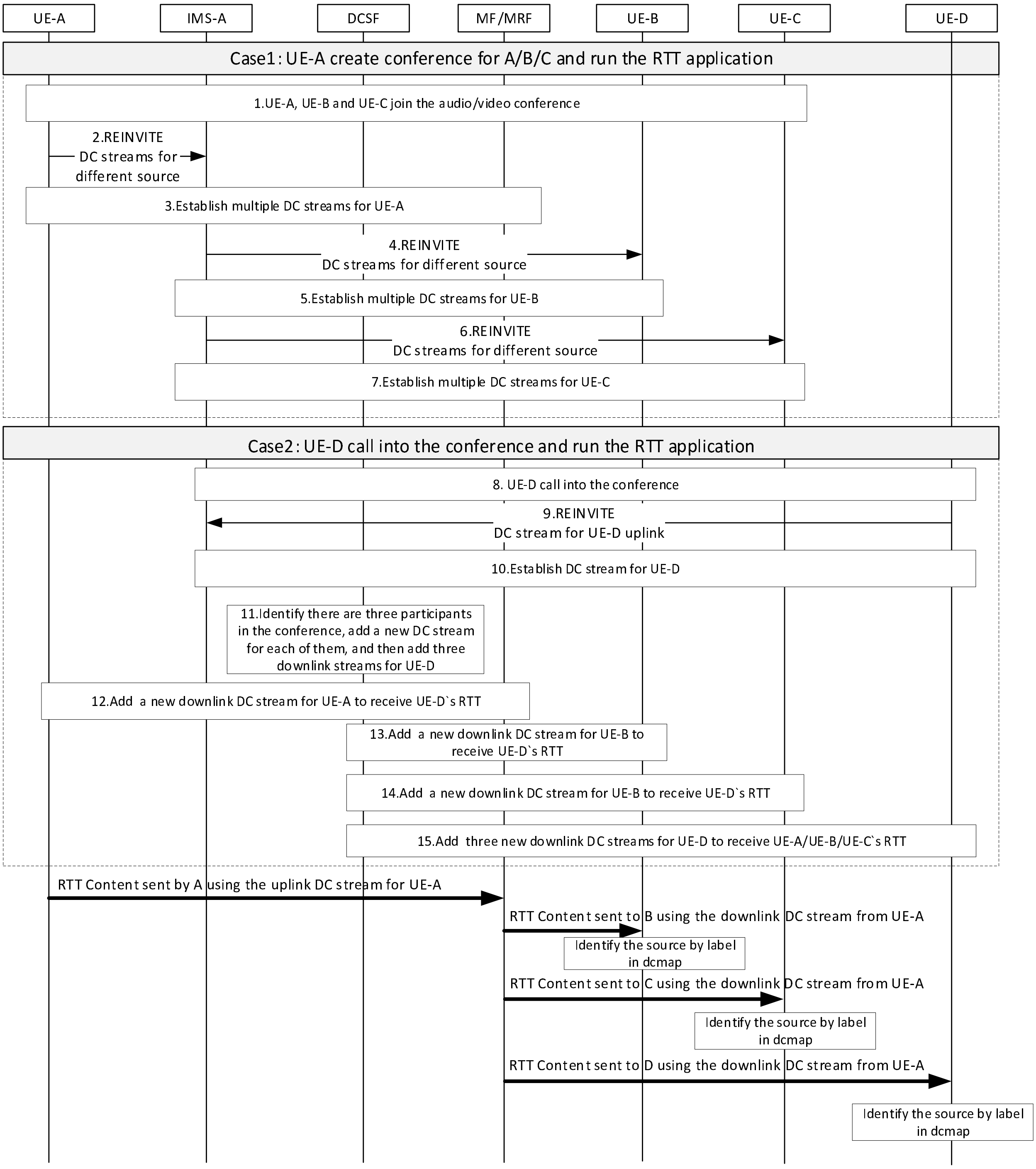

Figure 6.2.1.1-1 Multi DC Streams with UE Aware Mode Call Flow

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:

Table 6.2.1.1.1: SDP example

SDP 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=setup:actpass a=dcmap:1200 label="A-Identity";subprotocol="t140" a=dcsa:1200 fmtp:t140 cps=30 sendonly a=dcsa:1200 hlang-send:es eo a=dcmap:1201 label="B-Identity";subprotocol="t140" a=dcsa:1201 fmtp:t140 cps=30 recvonly a=dcsa:1201 hlang-recv:es eo a=dcmap:1202 label="C-Identity";subprotocol="t140" a=dcsa:1202 fmtp:t140 cps=30 recvonly a=dcsa:1202 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, MF/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.

6.2.1.2 UE Unaware Mode

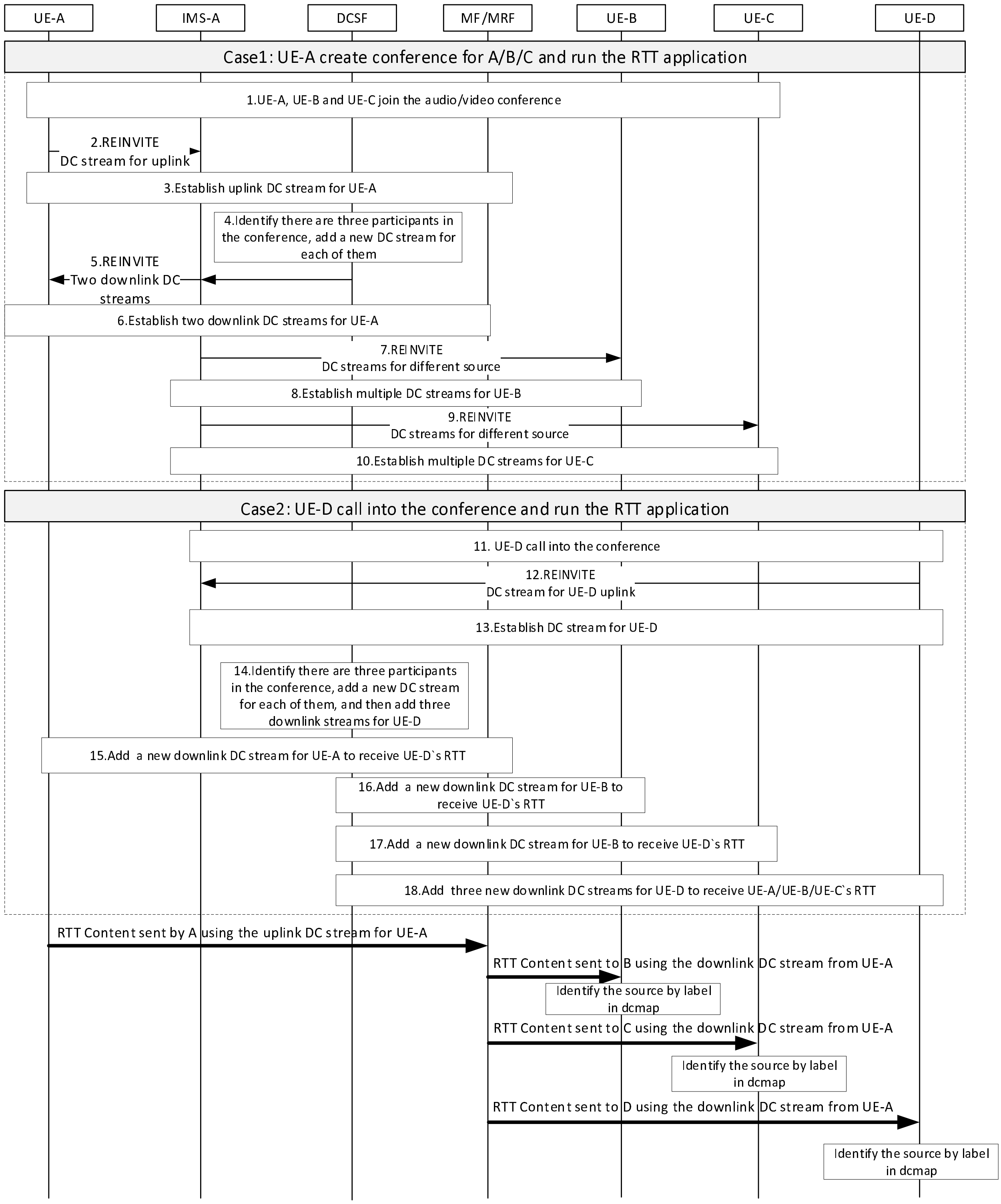

Figure 6.2.1.2-1 Multi DC Streams with UE Unaware Mode Call Flow

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:

Table 6.2.1.2.1: SDP example

SDP 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=setup:actpass a=dcmap:1200 label="A-Identity";subprotocol="t140" a=dcsa:1200 fmtp:t140 cps=30 sendonly a=dcsa:1200 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:

Table 6.2.1.2.2: SDP example

SDP 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=setup:actpass a=dcmap:1200 label="A-Identity";subprotocol="t140" a=dcsa:1200 fmtp:t140 cps=30 recvonly a=dcsa:1200 hlang-send:es eo a=dcmap:1201 label="B-Identity";subprotocol="t140" a=dcsa:1201 fmtp:t140 cps=30 sendonly a=dcsa:1201 hlang-recv:es eo a=dcmap:1202 label="C-Identity";subprotocol="t140" a=dcsa:1202 fmtp:t140 cps=30 sendonly a=dcsa:1202 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.

Annex <A> (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2023-11	SA4#126	S4-231625				Initial draft	0.1.0
2024-01	SA4#127	S4-240355				Revised and agreed	0.2.0
2024-03	SA#103	SP-240029				Version 1.0.0 created by MCC	1.0.0
2024-03						Version 18.0.0 created by MCC	18.0.0

History

Document history		
V18.0.0	May 2024	Publication