**Source: China Mobile Com. Corporation**

**Title: Multiparty RTT architecture and call flow for RTP solution**

## Document for: Discussion and Agreement

## Agenda Item: 10.9

# 1 Introduction

The MP\_RTT work item has been approved at the SA plenary #98-e in document SP-221346, and the use cases and requirements were agreed and incorporated in the PD v0.1.1 at 3GPP SA4-e (AH) RTC SWG post 122 in document S4aR230061.

This contribution proposes an architecture and call flow over RTP for the MP\_RTT.

# 2 Multi-party RTT Solutions

## 2.1 Multi-party RTT over RTP Solution

### 2.1.1 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.

### 2.1.2 Call Flow

#### 2.1.2.1 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.

#### 2.1.2.2 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 Proposal

We propose to agree to incorporate the architecture and the call flow into the MP\_RTT PD.

# References

1. TS 23.228: “IP Multimedia Subsystem (IMS)”
2. RFC9071: “RTP-Mixer Formatting of Multiparty Real-Time Text”
3. RFC4103: “RTP Payload for Text Conversation”