**3GPP TSG-CT WG1 Meeting #137-eC1-225014**

**E-Meeting, 18th – 26th August 2022 *was* C1-225014**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  |  | **CR** | **0330** | **rev** | **1** | **Current version:** |  |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **X** | Radio Access Network |  | Core Network |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Plugtest issue 10.1.4 of Nov 2022: Handling of release queued floor request re-transmission | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Samsung | | | | | | | | | |
| ***Source to TSG:*** | C1 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | MCProtoc18 | | | | |  | ***Date:*** | | | 2022-08-18 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | F |  | | | | | ***Release:*** | | | Rel-18 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | After sending ‘Release queued floor request’ in "U: queued" state, the client enters into "U: pending Release" state. In “U: pending Release" state, if the RTP packets are received then re-transmission timer for release floor request is stopped and state transition to "U: has no permission". Due to this, no retransmissions occurs if the first release floor request is lost between the client and the server which leads to state machine to be out of sync in client and server, i.e. The client assumes that the queued floor request was released and the server assumes that the queued floor request was not released. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | In “U: pending Release" state, if the RTP packets are received then we don’t need to stop the re-transmission timer (i.e. T100 - Floor Release) and state should remain in “U: pending Release" state until the floor messages (such as Taken, Idle messages) are received.  Changes: Removed the stoping of the T100 - Floor Release timer and avoiding transition to "U: has no permission" from "U: pending Release" state.  Figure 6.2.4.1-1 is updated to indicated that state will not change if RTP packets are received during 'U: pending Release' state | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Client state machine will be out of sync if the first request to release the queued floor request is losted and later server sends a Floor Granted to the user after processing the queued floor request and then waits until some timer expires and user is not in state to talkback. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 6.2.4.1 and 6.2.4.6.6 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  |  | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  |  | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  |  | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | | Rev1:   * figure 6.2.4.1-1 is updated and cover page is updated | | | | | | | | |

**\*\*\*\*\* FIRST CHANGE \*\*\*\*\***

#### 6.2.4.1 General

The floor participant shall behave according to the state diagram and the state transitions specified in this clause.

Figure 6.2.4.1-1 shows the state diagram for 'Floor participant state transition diagram for basic operation'.



Figure 6.2.4.1-1: Floor participant state transition diagram for basic operation.

State details are explained in the following clauses.

If an RTP media packet or a floor control message arrives in a state where there is no specific procedure specified for the RTP media packets or the received floor control message, the floor participant shall discard the floor control message or the RTP media packet and shall remain in the current state.

NOTE 1: A badly formatted RTP packet or floor control message received in any state is ignored by the floor participant and does not cause any change of the current state.

NOTE 2: The state transition diagram is the same for groups configured for audio cut-in floor control but the U: queued state should never be visited.

**\*\*\*\*\* NEXT CHANGE \*\*\*\*\***

##### 6.2.4.6.6 Receive RTP media packets (R: RTP Media)

Upon receiving an indication from the MCPTT client that RTP media packets are received, the floor participant:

1. shall request the MCPTT client to start rendering the RTP media packets;

2. should start the optional timer T103 (End of RTP media) for the participant from which the RTP packets were received;

3. void; and

4. shall remain in the 'U: pending Release' state.

NOTE: RTP media packets can be received from multiple sources when dual floor control is applied by the floor control server (see clause 6.3.6) or when multi-talker control is applied by the floor control server. The MCPTT client can differentiate between the different sources using the SSRC in the received RTP media packets. How the media mixer in the MCPTT client mixes the different RTP media stream sources is out of scope of the present document.

**\*\*\*\*\* END CHANGES \*\*\*\*\***