

TS.11- Annex N

Detailed Test Procedures for Mission Critical services Version 45

8th January 2024

*This is a Non-binding Permanent Reference Document of the GSMA*

Security Classification: Non-confidential – Full, Associate & Rapporteur Members

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Copyright © 2024 GSM Association

Disclaimer

The GSM Association (“Association”) makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Compliance Notice

The information contain herein is in full compliance with the GSM Association’s antitrust compliance policy.

This Permanent Reference Document is classified by GSMA as an Industry Specification, as such it has been developed and is maintained by GSMA in accordance with the provisions set out in GSMA AA.35 - Procedures for Industry Specifications.

Table of Contents

Annex N: Detailed Test Procedures for Mission Critical services 5

110 Mission Critical Push To Talk (MCPTT) 5

110.1 Registration 5

110.1.1 Mission Critical Bearer Setup 5

110.1.2 User Authentication 8

110.1.3 Service Authorisation 10

110.1.4 Key Retrieval 11

110.1.5 User profile and group configuration 13

110.2 Group Call, pre-arranged, on-demand 15

110.2.1 Normal Group Call 15

110.2.2 Emergency Group Call 20

110.2.3 Imminent Peril Group Call 25

110.2.4 Broadcast Call 31

110.2.5 Upgrade of call in progress to emergency or imminent peril call 36

110.2.6 Floor Control 39

110.2.6.1 On-Network Floor Control - Talk/Media Burst - when floor is Idle 39

110.2.6.2 On-Network Floor Control requested whilst floor is taken, and Queuing is not applied 43

110.2.7 Priorities 48

110.3 Group Call, chat, on-demand and pre-established session 49

110.3.1 Chat Group Call 49

110.4 Private Call, on-demand and pre-established session 55

110.4.1 Automatic commencement mode 55

110.4.2 Manual commencement mode 60

110.5 Affiliation 65

110.5.1 Affiliation 65

110.6 Client location submission 67

110.6.1 Client location submission 67

110.7 Ambience Listening Call 69

110.7.1 Remotely Initiated Ambience Listening 69

111 Mission Critical Data (MCData) 73

111.1 SDS (Short Data Service), standalone, over SIP 74

111.1.1 One-to-one SDS, standalone, over SIP 74

111.1.2 Group SDS, standalone, over SIP 76

111.2 Enhanced Status 79

111.2.1 Enhanced Status 79

112 Mission Critical Video (MCVideo) 81

**Annex H Glossary**

*Editor’s note: The following terms and abbreviations should be added to Annex H:*

|  |  |
| --- | --- |
| **Term** | **Description** |
| CAS/PAS | Controlling / Participating (MCPTT) Server |
| CF | Controlling Function (MCPTT) Server |
| CMS | Configuration Management Server |
| CSC | Common Service Core |
| CSCF | Call Server Control Function |
| DUT/DUTs | Device or Devices Under Test |
| Entity | An Entity is reference to a MCS User and MC UE |
| GMS | Group Management Server |
| GMK | Group Management Key |
| IdM Client/IdMC | Identity Management Client |
| IdMS | Identity Management Server |
| IMPU | IP Multimedia PUblic Identity |
| IMS | IP Multimedia Subsystem |
| K-ID | Key Identifier |
| KMS | Key Management Server |
| LTE/4G | Long Term Evolution - this is a reference to the the 4th Generation of cellular technology |
| MCS | Mission Critical System |
| MCX | Mission Critical Services – where the X refers to Push to Talk, Video or Data |
| MCX UE | Mission Critical Service User Equipment |
| MCPTT | Mission Critical Push to Talk |
| MCData | Mission Critical Data |
| MCVideo | Mission Critical Video |
| MEA | MCPTT Emergency Alert state |
| MEG | MCPTT Emergency Group state |
| MEGC | MCPTT Emergency Group Call state |
| MIG | MCPTT Imminent Peril Group state |
| MIGC | MCPTT Imminent Peril Group Call state |
| MIKEY-SAKKE | Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (see RFC 6509) |
| O-PF | Originating Participating Function (MCPTT) Server |
| PTT | Push To Talk |
| P-CSCF | Proxy Call Server Control Function |
| RTP | Real-time Transport Protocol |
| S-CSCF | Serving Call Server Control Function |
| SIP | Session Initiated Protocol |
| T-PF | Terminating Participating Function (MCPTT) Server |
| URI | Uniform Resource Identifier |

# Annex N: Detailed Test Procedures for Mission Critical services

This Annex contains the detailed procedures that are recommended to be used for Field and Lab Tests of mission critical services.

To validate the mission critical test cases in this annex, it is necessary to have access to configuration and trace log data from the device under test (DUT), as well as interface protocol data including SIP and PCAP protocol traces,as a minimum set for test environment.

However, it is strongly recommended, for some test cases necessary, to have access to the configuration settings and traces from the server side, such as the MCPTT server, Group Management Server, and other relevant servers. In some cases, this data may not be accessible (e.g. for security concerns in mission critical networks). If this is the case, the test cases will need to be verified based on alternative verification methods, e.g. human observations of the expected behaviour, as specified in the 3GPP standards.

The server side is not the implementation under test, but it is worth noting that there may be variations in how a test case is verified, depending on the specific implementation, e.g. the user interface is not in scope of the 3GPP standards and a notification can be implemnted or confgured to be either by audio or visual indicator or other types of notification.

# 110 Mission Critical Push To Talk (MCPTT)

## 110.1 Registration

### 110.1.1 Mission Critical Bearer Setup

**Description**

The purpose of this test is to validate the successful mission critical bearer setup before user authentication. The QCI value of a mission critical bearer is 69. Depending on the network design, this bearer can be configured to be set up in two ways: 1. As default bearer or 2. As a persistent dedicated bearer in which case another bearer is set up beforehand as default bearer.

**Applicability**

3GPP MCS Rel. 15 or later

**Related core specifications**

3GPP TS 23.401

3GPP TS 36.331

**Reason for the test**

To verify that a device under test (DUT) has set up the mission critical bearer and other bearer(s), if required, to carry on with mission critical set up signalling.

**Initial configuration**

Components:

* DUT: a mobile device with MCX Client App,
* MS core, Network and Cell that support mission critical bearer
* DUT has a MCX capable application
* DUT is switched off
* DUT has valid SIM with mission critical services user profile

**Test procedure**

**Scenario A: mission critical bearer is default bearer**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Switch on (power on) DUT. | DUT successfully establishes RRC connection.  “RRC connection setup complete” message is seen.  **VERIFICATION METHOD**:   * DUT air interface (Uu) log traces |
| 2 | DUT attaches to the network | DUT sends “Attach request” and “PDN connectivity request” to the network.  **VERIFICATION METHOD**:   * DUT air interface (Uu) log traces |
| 3 | DUT activates the default bearer | DUT receives “Attach accept” and “Activate default EPS bearer context request “.  Verify the value of “Quality of Service Class Identifier (QCI)” to be 69.  DUT sends “Attach Complete” and “Activate default EPS bearer context accept”.  At this point, mission critical default bearer is established.  **VERIFICATION METHOD**:   * DUT air interface (Uu) log traces |

**Example message flow for mission critical bearer setup for scenario A**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Mission critical bearer setup | | | |
| 1 | --> | DUT sends “Attach request” and “PDN connectivity request” | Only NAS messages are given |
| 2 | <-- | DUT receives “Attach accept” and “Activate default EPS bearer context request” | Verify QCI value in “Activate default EPS bearer context request” to be 69 |
| 3 | --> | DUT sends “Attach Complete” and “Activate default EPS bearer context accept”. | By this time DUT will have activated the default bearer. Completion of attach means default bearer which is also mission critical bearer is established |

**Scenario B: Mission critical bearer is a dedicated bearer**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Switch on (power on) DUT. | DUT successfully establishes RRC connection.  “RRC Connection Setup Complete” message is seen.  **VERIFICATION METHOD**:   * DUT air interface (Uu) log traces |
| 2 | DUT attaches to the network | DUT sends “Attach request” and “PDN connectivity request” to the network.  **VERIFICATION METHOD**:   * DUT air interface (Uu) log traces |
| 3 | DUT activates the default bearer | DUT receives “Attach accept” and “Activate default EPS bearer context request “.  Verify the value of “Quality of Service Class Identifier (QCI)” to be another value than 69.  DUT sends “Attach Complete” and “Activate default EPS bearer context accept”.  At this point, default bearer is established.  **VERIFICATION METHOD**:   * DUT air interface (Uu) log traces |
| 4 | DUT activates dedicated critical bearer | DUT receives “Activate dedicated EPS bearer context request”.  Verify the value of “Quality of Service Class Identifier (QCI)” to be 69.  DUT sends “Activate dedicated EPS bearer context accept”.  At this point, mission critical bearer is established.  **VERIFICATION METHOD**:   * DUT air interface (Uu) log traces |

**Example message flow for mission critical bearer setup for scenaio B**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Mission critical bearer setup | | | |
| 1 | --> | DUT sends “Attach request” and “PDN connectivity request”. | Only NAS messages are given |
| 2 | <-- | DUT receives “Attach accept” and “Activate default EPS bearer context request”. | Verify QCI value in “Activate default EPS bearer context request” noty to be 69 |
| 3 | --> | DUT sends “Attach Complete” and “Activate default EPS bearer context accept”. | By this time DUT will have activated the default bearer |
| 4 | <-- | DUT receives “Activate dedicated EPS bearer context request”. | Verify the value of “Quality of Service Class Identifier (QCI)” to be 69. |
| 5 | --> | DUT sends “Activate dedicated EPS bearer context accept”. | At this point, mission critical bearer is established. |

### 110.1.2 User Authentication

**Description**

The purpose of this test is to validate the successful user authentication including a successful MCX (using PTT placeholder for DATA, Video) Client registration:

**Applicability**

3GPP MCS Rel. 15 or later

**Related core specifications**

3GPP TS 33.102, 5.1.2

3GPP TS 33.401, 6.1

3GPP TS 33.179

3GPP TS 33.180

**Reason for test**

To verify that device under test (DUT) successfully initiates user authentication and network authentication service registration. It is assumed that entity authentication should occur at each connection set-up between the user and the network.

MCX Client switches on, can access IDMS and run authentication mechanism successfully

**Initial configuration**

Components:

* DUT: mobile device with Client App
* IdMS
* MS core, Network and Cell
* DUT has a MCX capable application
* DUT is switched off
* DUT has valid SIM for test network
* Minimum is that connectivity should be established between all elements of the specific scenario (and tested prior this case)
* The IdMS shall have the MC User, Service and Credential pre-provisioned and authentication must be performed over the CSC-1 reference point (between the IdM Client and IdM Server)
* (The CSC-1 shall consist of two endpoints; the authorization endpoint and token endpoint.)
* Authentication is performed using Open Id Connect 1.0

**Test procedure**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Power on DUT and confirm successful attach procedure. | DUT registers – after contacting the network to get the subscriber data and authentication data – successfully on Cell of a (home) network.  **VERIFICATION METHOD**:   * Log traces on DUT or IMS |
| 2 | Open MCX Client Application. | Authentication Request will appear  Network service is indicated on the display on DUT with MCX Client Application  **VERIFICATION METHOD**:   * Visual verification on GUI (of DUT)) |
| 3 | Log into the MCX Application using a valid account (Username and Password). | MCX User successfully authenticates (and is authorized to use its MCX services)   * that includes token is requested by MCX client * User is verified and data and identity token are correctly received according 3GPP TS 24 482 * Following Service Authorisation test case: functions available with the user (rights) are appearing  (to be defined)   Background activities coming with this test on IMS Core: MCPTT Client Registers & S-CSCF send 3rd party registration, after MCPTT User has already authenticated with IdMS  **VERIFICATION METHOD**:   * Input on GUI of DUT, and * Log traces on DUT or IMS |

**Example message flow for MCX User authentication to IdMS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| User Registration | | | |
| 1 | --> | MCX User send authorisation request to IdMS |  |
| 2 | <-- | IdMS redirects Authorisation mechanism |  |
| 3 | --> | MCX User sends Token Request |  |
| 4 | --> | IdMS sends Token Response |  |

### 110.1.3 Service Authorisation

**Description**

The purpose of this test is to validate the successful service registration and authorisation.

**Applicability**

3GPP MCS Rel. 15 or later

**Related core specifications**

3GPP TS 33.102, 5.1.2

3GPP TS 33.401, 6.1

3GPP TS 33.179

3GPP TS 33.180

**Reason for test**

To verify the successful execution of at least one of the 3 following mechanisms for service registration and authorisation.

1. MCX User authorised via IdMS
2. MCX User authorised via 3rd party registration
3. MCX User authorised using PUBLISH mechanism

**Initial configuration**

DUT: mobile device with MCX client App

IdMS

For scenario b and c need IMS core, and transportation network and Cell

MCX Client has valid SIM in network (if network is needed), minimum is that IP Connectivity should be established between all elements of the specific scenario, e.g. Access of IdMS via APN and tunnel mechanism

For c SIP PUBLISH based service authorization mechanism must be available

**Test procedure**

**Scenario A: Service Authorisation via IdMS** (Preconditions: access and identity tokens are available)

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Access IdMS (CMS URL; hardcoded) and proceed. | Proceed (Crossref) User Authentication – Service allocation is inherited.  **VERIFICATION METHOD**:   * Log traces |
| 2 | Open MCX Client Application. | Authentication Request will appear  Network service is indicated on the display on UE with MCX Client Application  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |

**Scenario B: Service Authorisation via 3rd party registration**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Register with IMPU and MCX (PTT) specific information mcptt-Info | * REGISTER to P-CSCF * REGISTER to S-CSCF * S-CSCF creates 3rd party Register + original REGISTER as body   **VERIFICATION METHOD:**   * Log traces on DUT or * Log traces on P-CSCF/S-CSCF |
| 2 | User registers to IMS Core and MCX Participating (IMPU or mcptt\_id binding). | Service authorisation completed (visible on device)  **VERIFICATION METHOD**:   * Log traces on DUT, and * Visual verification on GUI of DUT |

**Scenario C: Service Authorisation via PUBLISH mechanism**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Register with IMPU and MCX (PTT) specific information mcptt-Info | * REGISTER to P-CSCF * REGISTER to S-CSCF * S-CSCF creates 3rd party Register + original REGISTER as body   **VERIFICATION METHOD**:   * Log traces on DUT, or * Log traces on P-CSCF/S-CSCF |
| 2 | After successful user authentication at IdMS, check that the MCX Client sends PUBLISH including poc-settings and mcptt\_info with the credentials. | * User registers to IMS Core and MCX Participating (IMPU or mcptt\_id binding) * Service authorisation completed (visible on device)   **VERIFICATION METHOD**:   * Log traces on DUT, or * Visual verification on GUI of DUT |

### 110.1.4 Key Retrieval

**Description**

In order to derive the keys, the MCS client will need to download specific key material from a Key Management Server (KMS). This test case deals with the standard procedure for this key material download from KMS to the MCS client. The MCS client will need to download both the KMS certificate and its own Key Set from the KMS.

Key Set information can be protected for confidentiality and integrity using a Transport key (Trk). How the Trk is exchanged between the MCS client and the KMS is out of scope of the present release of the MCS standard.

Two deployment options have been proposed by the MCS standard: a direct connection between the MCS client and the KMS or a non-direct connection that uses an HTTPS proxy in the path. The expected behaviour at the client side is the same for both direct and non-direct connections.

The MCS client will first need to obtain an access-token to be authenticated by the KMS (for example by using the IdMS server).

**Applicability**

3GPP MCS Rel. 15 or later

**Related core specifications**

3GPP TS 33.179

3GPP TS 33.180 5.3.3

**Reason for test**

To verify that MCPTT client has downloaded KMS certificate and its key material

**Initial configuration**

* DUT: mobile device with MCX client App
* IdMS
* KMS
* IMS/SIP core, and transportation network and Cell

**Pre-Conditions/ Configuration**

MCS Client has valid SIM in network (if network is needed), minimum is that IP Connectivity should be established between all elements of the specific scenario, e.g. Access of IdMS via APN and tunnel mechanism.

MCS client has done a successful user authentication and a secure connection has been established.

**Test procedure**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | MCS Client sends a HTTPS POST request to obtain the KMS certificate | HTTPS/1.1 POST request contains the access token and a KMS request  **VERIFICATION METHOD**:   * DUT MCS client logs |
| 2 | MCS Client receives KMS’s own certificate | HTTPS/1.1 200 OK contains KMS certificate  **VERIFICATION METHOD**:   * DUT MCS client logs |
| 3 | MCS Client sends a HTTPS POST request to obtain its key material | HTTPS/1.1 POST request contains the access token and a KMS request  **VERIFICATION METHOD**:   * DUT MCS client logs |
| 4 | MCS Client receives its key material | HTTPS the/1.1 200 OK contains its key material  **VERIFICATION METHOD**:   * DUT MCS client logs |

**Example message flow for key retrieval**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Mission critical bearer setup | | | |
| 1 | --> | DUT sends HTTPS/1.1 POST | HTTPS/1.1 POST contains access token and a KMS request |
| 2 | <-- | DUT receives HTTPS/1.1 200 OK | HTTPS/1.1 200 OK contains KMS certificate |
| 3 | --> | DUT sends HTTPS/1.1 POST | HTTPS/1.1 POST request contains the access token and a KMS request |
| 4 | <-- | DUT receives HTTPS/1.1 200 OK | HTTPS the/1.1 200 OK contains its key material |

### 110.1.5 User profile and group configuration

**Description**

This test case describes how MCS (user profile and group) configuration documents are retrieved online.

**Applicability**

3GPP MCS Rel. 15 or later

**Related core specifications**

3GPP TS 24.481

**Reason for test**

To verify the successful configuration of user profile and group subscriptions

**Initial configuration**

* DUT: mobile device with MCS client App
* CMS and GMS
* IMS/SIP core, and transportation network and Cell

**Pre-Conditions/ Configuration**

MCS Client has valid SIM in network (if network is needed), minimum is that IP Connectivity should be established between all elements of the specific scenario.

MCS UE has obtained the Application Unique identity (AUID) during MC user authentication and also obtained the addresses of the CMS and the GMS or their subscription proxy function public service identity.

**Test procedure**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | MCS client subscribes to user profile document at Configuration Management Server (CMS) or at subscription proxy function | Subscribe message contains AUID and the requested document.  **VERIFICATION METHOD**:   * DUT MCS client logs |
| 2 | MCS client receives notify message if there is update to the requested documents | Notify messages contains xcap-diff document with the URLs of the new/updated user profile document and ETAGs.  **VERIFICATION METHOD**:   * DUT MCS client logs |
| 3 | MCS client sends a HTTPS GET request to user profile URL | **VERIFICATION METHOD**:   * DUT MCS client logs |
| 4 | MCS client downloads the correct user profile document | MCS client is correctly configured according to the new/updated document.  **VERIFICATION METHOD**:   * DUT MCS client logs |
| 5 | MCS client subscribes to group configuration at Group Management Server (GMS) or at subscription proxy function | Subscribe message contains the group IDs  **VERIFICATION METHOD**:   * DUT MCS client logs |
| 6 | MCS client receives notify messages if there is update to the requested documents | Notify messages contains xcap-diff document with the URLs of the new/updated user profile document and ETAGs  **VERIFICATION METHOD**:   * DUT MCS client logs |
| 7 | MCS client sends a HTTPS GET request to the different GMS XDM related URLs | **VERIFICATION METHOD**:   * DUT MCS client logs |
| 8 | MCS client downloads the correct group configuration | MCS client is correctly configured according to the new/updated documents  **VERIFICATION METHOD**:   * DUT MCS client logs |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| User profile and group configuration | | | |
| 1 | --> | MCS client sends SIP SUBSCRIBE that contains the proper AUID |  |
| 2 | <-- | MCS client receives SIP NOTIFY with the identity of the new/updated user profile document URL |  |
| 3 | --> | MCS client sends a HTTPS/1.1 GET request to the user-profile of the default user’s URL |  |
| 4 | <-- | MCS client correctly downloads the user profile document | MCS client is correctly configure according the new/updated documentation |
| 5 | --> | MCS client sends SIP SUBSCRIBE that contains the proper GMS subscription function address |  |
| 6 | <-- | MCS client receives SIP NOTIFY with the identity of the new/updated group configurations with URLs of the related documents and ETAGs |  |
| 7 | --> | MCS client sends a HTTPS/1.1 GET request to the URL(s) of the new/updated group document(s) |  |
| 8 | <-- | MCS client correctly downloads the group document and receives HTTPS/1.1 200 OK | MCS client is correctly configure according the new/updated documentation |

## 110.2 Group Call, pre-arranged, on-demand

### 110.2.1 Normal Group Call

**Description**

The DUT (with MCX client application) shall successfully perform the “Normal Group Call” and “Termination of Normal Group Call” procedures to MCX server.

**Applicability**

3GPP Rel. 15 or later

**Related core specifications**

3GPP TS 33.180

3GPP TS 23.379

3GPP TS 24.379

**Reason for test**

To verify the DUT can successfully establish Normal Group Call between different users and Termination of Normal Group Call

**Initial configuration**

The DUT, Client B and Client C MCX devices are assigned to a common group.

All devices have successfully performed User Authentication and been assigned a valid Token.

The DUT is authorised to initiate a Group call.

There are no group Emergency or Imminent Peril calls in progress.

**Test procedure**

**Scenario A: Normal Group Call**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Select a PTT group. | The group can be selected  **VERIFICATION METHOD:**   * Visual verification on GUI of DUT |
| 2 | Press the PTT button to initiate a group call. | Group Call request → MCPTT Server  OPTIONAL: Resolve MCPTT Group X ID→ Group Management Server  MCPTT Server → Group Call request to MCPTT Client B MCPTT Server → Group Call request to MCPTT Client C Waiting for permission to start a group call  MCPTT Client B → Group Call response  MCPTT Client C → Group Call response  Group management Server grants access to group X and the group call starts  Media plane and floor control establishment  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 3 | Check that an audio connection has been established with the PTT group. | The voice connection works to B and C  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-B and DUT-C * Audio sent by DUT-A should be audible on DUT-B and C. MC-PTT User ID of DUT-A user should be visible on DUT-B and DUT-C. |
| 4 | Release the PTT button to release the group call. | MCPTT Server: Generate group call release  Group Call release request to MCPTT Client B and MCPTT Client C (Notify user)  Group Call release response of MCPTT Client B and MCPTT Client C  Releasing floor control and media plane resources associated with group call  The group call ends  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-B and DUT-C:   + No Audio audible anymore on DUT-B and DUT-C.   + Active talkburst indication should be cleared on DUT-B and DUT-C. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Normal Group Call | | | |
| 1 | --> | DUT-A send group INVITE to the server |  |
| 2 | <-- | The server send INVITE to DUT-B and DUT-C |  |
| 3 | --> | DUT-B send ringing and OK to the server |  |
| 4 | --> | DUT-C send ringing and OK to the server |  |
| 5 | <-- | The server send OK to DUT-A |  |
| 6 | --> | A send audio to the server |  |
| 7 | <-- | The server send audio to DUT-B and DUT-C |  |
| Termination of Normal Group Call | | | |
| 8 | --> | DUT-A send BYE to server |  |
| 9 | <-- | Server send OK to DUT-A |  |
| 10 | <-- | Server send Disconnect to DUT-B and DUT-C |  |
| 11 | --> | DUT-B send ACK to server |  |
| 12 | --> | DUT-C send ACK to server |  |

**Scenario B: Normal Group Call (time out)**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Select a PTT group. | The group can be selected  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 2 | Press the PTT button to initiate a group call. | Waiting for permission to start a group call  Server grants access to a group and the group call starts  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 3 | Check that an audio connection has been established with the PTT group. | The voice connection works to B and C  **VERIFICATION METHOD**:   * Audio and Visual verification on DUT-B and DUT-C: GUI/Speaker * Audio sent by DUT-A should be audible on DUT-B and DUT-C. MC-PTT User ID of DUT-A user should be visible on DUT-B and DUT-C. |
| 4 | Speech item time out. | The group call ends  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 5 | Press PTT button on DUT-B | The voice connection works to DUT-A and DUT-C  **VERIFICATION METHOD**:   * Audio and Visual verificationl |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Normal Group Call | | | |
| 1 | --> | DUT-A send group INVITE to the server |  |
| 2 | <-- | The server send INVITE to DUT-B and DUT-C |  |
| 3 | --> | DUT-B send ringing and OK to the server |  |
| 4 | --> | DUT-C send ringing and OK to the server |  |
| 5 | <-- | The server send OK to DUT-A |  |
| 6 | --> | A send audio to the server |  |
| 7 | <-- | The server send audio to DUT-B and DUT-C |  |
| Termination of Normal Group Call | | | |
| 8 | --> | Server send Disconnect to DUT-A, DUT-B and DUT-C |  |
| 10 | --> | DUT-A send ACK to server |  |
| 11 | --> | DUT-B send ACK to server |  |
| 12 | --> | DUT-C send ACK to server |  |

**Scenario C: Normal Group Call (toggle mode)**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Select a PTT group. | The group can be selected  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 2 | Press the PTT button and release it. | Waiting for permission to start a group call  Server grants access to a group and the group call starts  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 3 | Check that an audio connection has been established with the PTT group. | The voice connection works to DUT-B and DUT-C  **VERIFICATION METHOD**:   * Audio and Visual verification on DUT-B and DUT-C: GUI/Speaker * Audio sent by DUT-A should be audible on DUT-B and DUT-C. MC-PTT User ID of DUT-A user should be visible on DUT-B and DUT-C. |
| 4 | Press the end-call button. | The group call ends  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-B and DUT-C. * Audio should not be audible anymore on DUT-B and DUT-C. Active talkburst indication should be cleared on DUT-B and DUT-C. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Normal Group Call | | | |
| 1 | --> | DUT-A send group INVITE to the server |  |
| 2 | <-- | The server send INVITE to DUT-B and DUT-C |  |
| 3 | --> | DUT-B send ringing and OK to the server |  |
| 4 | --> | DUT-C send ringing and OK to the server |  |
| 5 | <-- | The server send OK to DUT-A |  |
| 6 | --> | A send audio to the server |  |
| 7 | <-- | The server send audio to DUT-B and DUT-C |  |
| Termination of Normal Group Call | | | |
| 8 | --> | DUT-A send BYE to server |  |
| 9 | <-- | Server send OK to DUT-A |  |
| 10 | <-- | Server send Disconnect to DUT-B and DUT-C |  |
| 11 | --> | DUT-B send ACK to server |  |
| 12 | --> | DUT-C send ACK to server |  |

### 110.2.2 Emergency Group Call

**Description**

In this test case a MCPTT user initiates an MCPTT emergency group call with affiliated MCPTT members of that MCPTT group. The initiating MCPTT client gains elevated access privilege over other MCPTT clients and resources that are in normal state of operation. In this test case all affiliated MCPTT clients are in idle mode.

The initiation of a MCPTT emergency group call puts the MCPTT group into the in-progress emergency state.by the MCPTT server until the in-progress emergency state of the MCPTT group is cancelled.

This test case will require a minimum of three MCPTT clients which are all on the same MCPTT server and Group Management Server (GMS) and are affiliated, and active on the same MCPTT Group.

**Applicability**

3GPP MCS Rel.15 or later

**Related core specifications / References**

3GPP TS 23.379

3GPP TS 24.379

3GPP TS 24.481

**Reason for test**

To verify that the DUT can successfully establish a MCPTT Emergency Group Call and provide priority access to the requesting DUT over a Normal Group Call that may be in progress.

**Initial configuration**

The following pre-conditions are applied for this test case:

1. Three MCX DUTs are connected via an core network using the process as defined in 3GPP TS 33.401.
2. The MCX DUTs are all switched on and have been authorised/authenticated and assigned to a common group.
3. The Requesting DUT and Receiving DUTs are reachable on the Home KMS and Access Network.
4. Functions of the MCPTT server (Originating Participating/Controlling/Terminating Participating MCPTT Function) and all MCPTT DUTs are located in the primary MCPTT system.
5. A pre-arranged group is established with a common MCPTT group ID and member list on the same Group Management Server (GMS).
6. The MCPTT User Profile will have the following parameters set:
   1. Allow-emergency-group-call = true
7. The MCPTT service setting for ruleset actions Allow-Automatic-Commencement= false.
8. Initial configuration of each DUTs under test will be:
   1. MCPTT Emergency state = clear
   2. MCPTT Emergency Group (MEG) state = MEG 1: no-emergency
   3. MCPTT Emergency Group Call (MEGC) state = MEGC: emergency-gc-capable
   4. MCPTT Emergency Alert (MEA) state= no-alert
9. The Controlling MCPTT function characteristics will be:
   1. In-progress emergency group state = false
10. The DUT User Profile shall be identified as being authorized with the following configuration:
    1. allow-emergency-group-call = true
    2. MCPTTGroupInitiation = DedicatedGroup or UseCurrentlySelectedGroup
11. Three DUTs are all in the same MCPTT talkgroup used in this test case and each will have MCPTTGroupPriority Hierarchy as follows:
    1. DUT-A: MCPTTGroupPriority Hierarchy=low
    2. DUT-B: MCPTTGroupPriority Hierarchy=medium
    3. DUT-C: MCPTTGroupPriority Hierarchy=high

NOTE: The Group Priority Hierachy is a non-negative integer value between 0 and 8.

1. The minimumReportInterval timer is set to notify the current location when a DUT is in a MCPTT emergency group call

**Test Procedure**

**Scenario A: Emergency Group Call with Idle floor**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition | Floor is idle for all DUTs  **VERIFICATION METHOD:**   * Visual GUI or log traces on all DUTs |
| 2 | Emergency call is initiated on DUT-A | For DUT-A the MCPTT client checks if the user is authorized to initiate a MCPTT emergency group call if their profile has:  - <mcptt-client-id> shall be identity of DUT-A user  - <allow-emergency-group-call> = true, &  - MCPTTGroupInitiation = [DedicatedGroup or UserCurrentlySelectedGroup]    Following successful authorization for DUT-A the following status attributes value are set as follows:  - MCPTT emergency state= set  - MEG state= MEG 4: confirm-pending  MEGC state= MEGC 2: emergency-call-requested  **VERIFICATION METHOD**:   * Visual GUI or Log traces on DUT-A |
| 3 | DUT-A sends MCPTT emergency group call request to the MCPTT Server | DUT-A sends a SIP INVITE or REFER message to the MCPTT Server. This shall include in the application/vnd.3gpp.mcptt-info+xml MIME body the following:  - <emergency-ind> element set to “true”  - <alert-ind> element set to “false”  - MEA state= MEA 2: emergency-alert-confirm-pending  Shall include in the application/vnd.3gpp. mcptt-location-info+xml MIME body the current location information of DUT-A with the following details:  - <location-information> <Configuration> <EmergencyLocationInformation> <Report>  - <ReportID>  - <RequestID>  - <TriggerID>  - <CurrentLocation>  **VERIFICATION METHOD:**   * Log traces or Configuration on DUT-A, or * SIP traces on interface |
| 4 | MCPTT Server receives emergency group call request and determinesaffiliation status determination for DUT-A | On receiving the SIP INVITE/REFER, the MCPTT server, it will determine the affiliation status of DUT-A, by checking the following:  - MCPTT user information,  - MCPTT client information,  - MCPTT group information & affiliation status,  As defined in 3GPP TS 24.279 clause 9.2.2.2.11.  **VERIFICATION METHOD:**   * not verifiable on DUT / Log traces on server side |
| 5 | MCPTT Server checks if DUT-A user request is authorised | Check from the originating SIP message the following:  - <EmergencyAlert><entry><entry-info> set to “DedicatedGroup” or “UseCurrentlySelectedGroup”  For the MCPTT Group set:  <allow-emergency-group-call> value to “true”.  **VERIFICATION METHOD**:   * not verified on DUT, or * SIP traces on interface * Log traces on server side |
| 6 | MCPTT Server Verifies with the GMS if Emergency MCPTT Initiating DUT-A client is authorised | Query the GMS and check the ruleset for the following:   * common/ruleset/rule/id{i}/ actions/Allow-Emergency-Group-Call= true * common/MCPTT-group-call/EmergencyCall/MCPTTGroupInitiation/entry= DedicatedGroup * common/MCPTT-group-call/EmergencyCall/MCPTTGroupInitiation/entry/uri-entry = [TBD] * identify the active MCPTT Clients & DUTs affiliated with the initiating MCPTT Group.   The following attributes are configured for the MCPTT Group:   * in-progress emergency group (IPEG) state = true   [Ref: TS24.379 clause 6.3.2.1.8.1 (1)]  **VERIFICATION METHOD**:   * not verfiable on DUT, or * Log traces on server or GMS side |
| 7 | MCPTT Serververifies that auto commencement mode is set | Verify that common/ruleset/rule/id{i}/actions/ Allow-Automatic-Commencement=true  **VERIFICATION METHOD**:   * not verified on DUT / log traces on server side |
| 8 | MCPTT Server sends to DUT-B and DUT-C a SIP INVITE for emergency call | The SIP INVITE application/vnd.3gpp.mcptt-info+xml MIME body will contain the:   * <mcptt-request-uri> identifies the MCPTT ID of the terminating (DUT-B or DUT-C) * <mcptt-calling-user-id> identifies MCPTT ID of DUT-A * <mcptt-calling-group-id> identifies the MCPTT Group ID of the emergency group call * <emergency-ind> is set to “true” * <alert-ind> is set to “false”   Optionally if location details are supported the ‘application/vnd.3gpp.mcptt-location-info+xml MIME’ will contain the following:   * <location-info><Report> * <ReportType>Emergency</ReportType> * <Configuration><EmergencyLocationInformation> <CurrentLocation>   **VERIFICATION METHOD**:   * Log traces on DUT-B, DUT-C, or * SIP traces on interface |
| 9 | DUT-B and DUT-C validate and accept terminating MCPTT emergency group call SIP INVITE/REFER request | Check if terminating DUT-B or DUT-C is already in MCPTT emergency call or the maximum number of emergency calls are already in progress, if not then perform the following:    Display on terminating DUT-B & DUT-C the following:  - MCPTT ID of DUT-A  - MCPTT group ID with emergency condition    Set status on termination DUT-B & DUT-C as follows:  - MEG state= MEG 2: in-progress  - MIG state= MIG 1: no-imminent-peril  - MIGC state= MIGC 1: imminent-peril-gc-capable  **VERIFICATION METHOD**:   * Log traces on DUT-B, DUT-C |
| 10 | DUT-B and  DUT-C accept terminating MCPTT emergency call request | Terminating DUT-B & DUT-C accept MCPTT emergency group call by responding with a SIP 200 (OK) message.  **VERIFICATION METHOD**:   * + Audio and Visual verification on DUT-B and C: GUI/Speaker.Emergency group call should be indicated on DUT-B and DUT-CMCPTT ID of DUT-A should be displayed on DUT-B and DUT-C |
| 11 | MCPTT Server grants DUT-A the floor (by sending a SIP 200 (OK) response to DUT-A to accept originating MCPTT emergency call request) | Originating DUT-A MCPTT emergency group call is accepted and the following states are updated for the MCPTT user:  - MEG state= MEG 2: in-progress  - MEGC state: MEGC 3: emergency-call-granted  - MIGC state= MIGC 1: imminent-peril-capable  - MIG state= MIG 1: no-imminent-peril  **VERIFICATION METHOD**:   * Log Traces on DUT-B and DUT-C |
| 12 | DUT-A    Granted Floor Control | DUT-A is granted floor control and the MCPTT client will notify the MCPTT user by playing a Permission Initiation tone  **VERIFICATION METHOD**:   * Audio and/or visual verification on DUT-A |
| 13 | DUT-A à MCPTT Server  MCPTT Server à DUT-B &  MCPTT Server à DUT-C    RTP Voice Media plane established | DUT-A presses the PTT button to speak and a RTP Voice Media plane is established to all active and affiliated members of the MCPTT Emergency Group call.  **VERIFICATION METHOD**:   * Audio and/or Visual verification on DUT-B and DUT-C. * MCPTT-ID of DUT-A is displayed on DUT-B and DUT-C as speaking party. * Audio of DUT-A is audible on DUT-B and DUT-C |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT – NW** | **Message** | **Comments** |
| Emergency Group Call with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP INVITE for MCPTT emergency group call |  |
| 2 | <-- | MCPTT Server sends SIP INVITE to DUT-B for MCPTT emergency group call |  |
| 3 | <-- | MCPTT Server sends SIP INVITE to DUT-C for MCPTT emergency group call |  |
| 4 | --> | DUT-B sends SIP 200 (OK) response to MCPTT Server |  |
| 5 | --> | DUT-C sends SIP 200 (OK) response to MCPTT Server |  |
| 6 | <-- | MCPTT Server sends SIP 200 (OK) to DUT-A |  |
| 7 | --> | DUT-A sends audio to MCPTT Server |  |
| 8 | <-- | MCPTT Server sends audio to DUT-B & DUT-C |  |

**Details flow sequence diagram:**

A picture containing text, screenshot, diagram, parallel

Description automatically generated

**Scenario B: Emergency Group Call with extsing talkburst on Talkgroup**

### 110.2.3 Imminent Peril Group Call

**Description**

An imminent peril group call in MCPTT is initiated on-network when a member of an MCPTT group is in an imminent peril situation and requires assistance from other members affiliated with the same MCPTT group. In this test case, we have a minimum of three Devices Under Test (DUTs). The initiating member's device (DUT-A) sends an emergency alert to all other members of the same MCPTT group (DUT-B & DUT-C). This alert indicates that the initiating member is in an imminent peril situation and requires immediate assistance.

Upon receiving the emergency alert, all members affiliated with the same MCPTT group are notified of the call and can join the group call. In this test case, we have initiated the imminent peril call whilst the MCPTT group is in idle mode, The call is established as a high-priority call, which means it takes precedence over other ongoing calls in the network.

During the imminent peril group call, the initiating member can provide details about their situation and location to other members of the group. This enables other members to take appropriate action and provide assistance as needed. All members of the group call have the ability to communicate with each other in real time, which allows for coordinated response efforts.

The call remains active until all members of the group have confirmed that the situation has been resolved, or until the initiating member cancels the call. After the call, an incident report is generated, which includes details about the call, such as the initiating member's device information, call duration, and call participant information.

Overall, the MCPTT imminent peril group call provides a critical communication channel for members of the same MCPTT group to respond quickly and effectively to an imminent-peril threat situation.

The initiation of an MCPTT imminent peril group call puts the MCPTT group into the in-progress imminent peril state (NOTE: this state is not associated with the user but the group).by the MCPTT server until the in-progress imminent peril state of the MCPTT group is cancelled.

**Applicability**

3GPP MCS Rel.15 or later

**Related core specifications / References**

3GPP TS 23.379

3GPP TS 24.379

3GPP TS 24.481

3GPP TS 24.483

3GPP TS 24.484

**Reason for test**

The purpose of the test is to verify that the DUT can successfully establish a high-priority MCPTT Imminent Peril Group Call, which takes precedence over any normal group calls that may be in progress. This ensures that the initiating member of the Imminent Peril Group Call is able to quickly and efficiently communicate their emergency situation to other members of the same MCPTT group and receive timely assistance. By verifying the successful establishment of the MCPTT Imminent Peril Group Call and priority access, the test helps to ensure the reliability and effectiveness of the MCPTT system during imminent peril situations.

**Initial configuration**

The following pre-conditions are applied for this test case:

1. Three MCS DUTs are connected On-Network via the core network using the process as defined in 3GPP TS 33.401.
2. The MCS DUTs are all switched-on and are authorised/authenticated and assigned to a common group.
3. The Requesting DUT and Receiving DUTs are reachable on the Home Key Management Server (KMS) and Home Access Network.
4. To support an imminent peril group call, the MCPTT client needs to be configured as follows:
   1. The <Allow-Imminent-Peril-Call> element should be set to "true" to allow the initiation of an imminent peril call.
   2. The client should be configured with audio and visual indications to alert the user to the imminent peril state and to confirm that the call has been initiated successfully.
   3. The client should handle the imminent peril group call in a way that prioritizes it over any other ongoing calls, such as non-emergency group calls or individual calls.
5. To support an imminent peril group call, the MCPTT user profile needs to be configured as follows:
   1. <Authorised> = **true**.
   2. <Cancel> = **true**.
   3. <MCPTTGroupInitiation>
      1. <GroupId> = e.g “***GroupID123-ImPer***” - this is the default GroupID for imminent peril.
      2. <Display Name> is a human-readable name for the imminent peril Group ID.
      3. <Usage> = “**UseCurrentlySelectedGroup**”

***NOTE****: if <Usage> = “UseCurrentlySelectedGroup” then GroupID selected by the DUT and affiliated to the user is used in the communications; if <Usage> = “DedicatedGroup” or no affiliate GroupiD is specified, then GroupID in (c)(i) is used as the GroupID (with its respective Group Names).*

1. A pre-arranged group is established with a common MCPTT group ID and member list on the same Group Management Server (GMS).
2. The DUT, its Location Management Client configured shall be set up to support an Imminent peril call scenario.
3. Functions of the MCPTT server (Originating Participating/Controlling/Terminating Participating MCPTT Function) and all MCPTT DUTs are located in the primary MCPTT system.
4. The MCPTT service setting for ruleset actions Allow-Automatic-Commencement= false.
5. In an idle mode (where there are no ongoing calls), the MCPTT client will support the following internal states:
   1. <allow-imminent-peril-call> = true
   2. MCPTT imminent peril group (MIG) state = MIG 1 (no-imminent-peril)
   3. MCPTT imminent peril group call (MIGC) state = MIGC 1 (imminent-peril-gc-capable)
   4. MCPTT emergency group call state = clear

***NOTE****: The MCPTT emergency state for this test case is required to remain with its value set to “clear”, as the emergency state takes presence over the imminent peril group call.*

1. In an idle mode (where there are no ongoing calls), the controlling MCPTT server will for the MCPTT Group have the following states set:
   1. In-progress imminent peril group (IPIG) state = false
   2. In-progress emergency group state = false
2. Three DUTs are used in this test case and each will have MCPTTGroupPriority Hierarchy as follows:
   1. DUT-A: MCPTTGroupPriority Hierarchy=low
   2. DUT-B: MCPTTGroupPriority Hierarchy=medium
   3. DUT-C: MCPTTGroupPriority Hierarchy=high

NOTE: The Group Priority Hierachy is a non-negative integer value between 0 and 8.

1. The MCPTT Server will include for the affiliated MCPTT Group the following Imminent Peril configuration:
   1. <on-net-in-progress-imminent-peril-state-cancellation-timeout> for these test cases will be set to PT5M (5 minutes).

**Test Procedure**

**Scenario A: Emergency Group Call with Idle floor**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition in idle mode | **VISUAL OBSERVATION**:  Floor is idle for all DUTs  **MCPTT CLIENT/USER OBSERVATION**:  For each DUT the following status conditions are set:   * MIG = MIG 1 (no imminent peril) * MIGC = MIGC 1 (imminent peril-gc-capable) * MCPTT emergency state = clear   **MCPTT SERVER OBSERVATION**:  The controlling MCPTT function has its state set as follows:   * In-progress imminent peril group state - false   **VERIFICATION METHOD**:   * Log traces on DUT-A * Log traces on server side |
| 2 | DUT-A initiated an Immient Peril group call by pressing their Imminent Peril button on their | **VISUAL OBSERVATION**:   * DUT-A presses Imminent Peril Button   **MCPTT CLIENT/USER OBSERVATION**:  For DUT-A the MCPTT client validates they are authorized to initiate a MCPTT imminent peril group call if their profile has:   * <allow-imminent-peril-call> = true   Upon successful authorization for DUT-A the following is done:   1. If DUT-A is authorised the DUT-A MCPTT Group ID and Name to use in its communication are identified, as follows:  * If <ImminentPerilCall> <MCPTTGroupInitiation> = [UseCurrentlySelectedGroup] then the current selected MCPTT Group ID & Name are selected; or * if  <ImminentPerilCall> <MCPTTGroupInitiation> =[DedicatedGroup], then the default imminent group ID and name are used  1. For DUT-A its status is updated as follows:  * MIGC state = MIGC 2 (imminent-peril-call-requested) * MIG state= MIG 4 (confirm-pending)   **VERIFICATION METHOD**:   * Log traces on DUT-A |
| 3 | DUT-A (MCPTT client) sends an Imminent Peril Request to the MCPTT Server. | DUT-A requests the MCPTT server to upgrade the MCPTT group to an in-progress imminent peril state by sending an MCPTT imminent peril group call request. The request may contain an indication of an implicit floor request. If the imminent peril group call request includes an implicit floor request it may also include location information.  **INTERFACE OBSERVATION**:  The DUT-A sends a SIP INVITE request message to the MCPTT Server this shall contain the following:   1. In the <application/vnd.3gpp.mcptt-info+xml> MIME body with the following  * <mcptt-calling-user-id> identifies the MCPTT ID * <imminentperil-ind> set to "true"; * MCPTT client emergency group call state is set to MIG2 * MEA state= MEA 2: emergency-alert-confirm-pending * <mcptt-request-uri> identify the MCPTT group ID  1. In the <application/vnd.3gpp. mcptt-location-info+xml> MIME body contains the current location information of DUT-A with the following details:  * <location-information><Configuration> <EmergencyLocationInformation><Report>   + <ReportID>   + <RequestID>   + <TriggerID> * <CurrentLocation>  1. In the <application/vnd.3gpp.mcptt-info+xml> MIME body will contain the <mcptt-request-uri> identify the MCPTT group ID;   **VERIFICATION METHOD**:   * Log traces on DUT-A * Log traces on server side * SIP traces on interface |
| 4 | MCPTT Serverffiliates DUT-A to group | The MCPTT server implicitly affiliates DUT-A to the imminent peril group if the lenient is not already affiliated.  The MCPTT server maintains a list of affiliated MCPTT group information entries that include:   * MCPTT user information, * MCPTT client information, * MCPTT group information & affiliation status,   As defined in 3GPP TS 24.279 clause 9.2.2.3.6 & 9.2.2.2.8  **VERIFICATION METHOD**:   * Log traces on server side |
| 5 | MCPTT Server  resolve DUT-A to authorise imminent peril call | MCPTT server checks whether DUT-A is authorized for initiation of imminent peril group calls on the indicated MCPTT group, and if authorized, it resolves the MCPTT group ID to determine the members of that MCPTT group and their affiliation status, based on the information from group management server. The MCPTT server checks whether the provided functional alias, if present, can be used and has been activated for the user.  **VERIFICATION METHOD**:   * Log traces on server side |
| 6 | MCPTT Serverexecutes Bearer Priority Configuration | The MCPTT server configures the priority of the underlying bearers for all participants in the MCPTT group. NOTE 2: Successive calls during the in-progress imminent peril state will all receive the adjusted bearer priority.  For an imminent priority setup, the EPS bearer priority is adjusted for all participants when the in-progress imminent peril state of the group is set whether or not they themselves initiated an imminent peril group call. For unicast bearers, this is achieved by using the Resource-Priority header field as specified in IETF RFC 4412 with namespaces defined for use by MCPTT specified in IETF RFC 8101, and for MBMS bearers this is achieved by having the participating MCPTT function adjust the ARP (priority, PVI, PCI) and executing the Modify MBMS Bearer Procedure per 3GPP TS 29.468.  **VERIFICATION METHOD**:   * Log traces on server side * Bearer config on network |
| 7 | MCPTT Server requests mminent Peril group call requested to participating DUTs B and C | MCPTT server sends the imminent peril group call request towards the MCPTT clients of each of those affiliated MCPTT group members (DUT-B & DUT-C).  The request contains an indication of the in-progress imminent peril. If location information was included in the imminent peril group call request, the MCPTT server checks the privacy policy of the MCPTT user to decide if the location information of MCPTT client 1 can be provided to other users on the call (refer to Annex A.3 "Authorisation to provide location information to other MCPTT users on a call when talking").  **MCPTT SERVER OBSERVATION**:   * Specify an MCPTT Session ID for the imminent peril   **INTERFACE OBSERVATION**:  The MCPTT Server will send a SIP INFO request towards DUT-B & DUT-C, containing the following information:   1. MCPTT Session ID 2. Iin the <application/vnd.3gpp.mcptt-info+xml> MIME body, for following will be set:  * <imminentperil-ind> is set to “true” * <mcptt-calling-user-id> identifies the MCPTT ID  1. <mcptt-calling-user-id> identifies the MCPTT ID   **VERIFICATION METHOD**:   * Log traces on DUT-B & DUT-C * Log traces on server side * SIP traces on interface |
| 8 | DUT-B & DUT-C Notify of imminent peril call | MCPTT users are notified of the incoming imminent peril call. The functional alias of the group call initiating MCPTT user may be displayed.  **MCPTT CLIENT/USER OBSERVATION**:   1. For DUT-B & DUT-C set the following states:  * MIGC state = MIGC3 (imminent-peril-call-granted) * MIG state= MIG 2 (in-progress)   **VERIFICATION METHOD**:   * Log traces on DUT-B & DUT-C * Log traces on server side * SIP traces on interface |
| 9 | DUT-B and DUT-Csend Imminent peril group call response to MCPTT Server | The receiving MCPTT clients send the MCPTT imminent peril group call response to the MCPTT server to acknowledge the imminent peril call request. For a multicast call, these acknowledgements are not set.  **INTERFACE OBSERVATION**:  DUT-B & DUT-C shall send a SIP 200 (OK) response back to the SIP INFO request.  **VERIFICATION METHOD**:   * Log traces on DUT-B & DUT-C * Log traces on server side * SIP traces on interface |
| 10 | DUT-A receives imminent peril group call response by MCPTT server | The MCPTT server sends the MCPTT imminent peril group call response to the MCPTT user 1 to inform the successful imminent peril call establishment.  **MCPTT CLIENT/USER OBSERVATION**:   * Set MIGC = MIGC 3 (imminent-peril-call-granted) * Set MIG = MIG 2 (in-progress)   **INTERFACE OBSERVATION**:   * The DUT-A receives a SIP 2xx response and. * The SIP 2xx response message contains no Warning header fields with warning text containing <mcptt-warn-code> of “149”   **VERIFICATION METHOD**:   * Log traces on DUT-A * Log traces on server side * SIP traces on the interface   NOTE: this step can occur at any time following step 9, on the conditions to proceed with the imminent peril call. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Emergency Group Call with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP INVITE for MCPTT imminent peril group call |  |
| 2 | <-- | MCPTT Server sends SIP INFO to DUT-B for MCPTT imminent peril group call |  |
| 3 | <-- | MCPTT Server sends SIP INFO to DUT-C for MCPTT imminent peril group call |  |
| 4 | --> | DUT-B sends SIP 200 (OK) response to MCPTT Server |  |
| 5 | --> | DUT-C sends SIP 200 (OK) response to MCPTT Server |  |
| 6 | <-- | MCPTT Server sends SIP 2xx (OK) to DUT-A |  |

### 110.2.4 Broadcast Call

**Description**

In this test case, an MCPTT DUT will trigger a broadcast group call

A user initiates an MCPTT broadcast group call with affiliated MCPTT members of that MCPTT group. The initiating MCPTT client gains elevated access privilege over other MCPTT clients and resources that are in normal state of operation as Broadcast calls are assigned an ARP level that is higher than normal group calls but lower than Emergency calls.

In this test case all affiliated MCPTT clients are in idle mode.

When the MCPTT user initiates a broadcast group call, the MCPTT client:

1) in the case the prearranged group call is initiated on-demand, shall include in the application/vnd.3gpp.mcptt-info+xml MIME body the <broadcast-ind> element set to "true".

2) in the case the prearranged group call is initiated using a pre-established session, shall include in the application/vnd.3gpp.mcptt-info+xml MIME body in the "body" URI header field in the Refer-To header field the <broadcast-ind> element set to "true".

The handling of the call is basically the same as other Group Call but only the call originating MCPTT user is allowed to transmit media and if the media transmission from call originating MCPTT user is complete, the broadcast group call is released.

This test case will require a minimum of three MCPTT clients which are all on the same MCPTT server and Group Management Server (GMS) and are affiliated, and active on the same MCPTT Group.

**Applicability**

3GPP MCS Rel.15 or later

**Related core specifications / References**

3GPP TS 23.379

3GPP TS 24.379

3GPP TS 24.481

**Reason for test**

To verify that the DUT can successfully establish and terminate an MCPTT Broadcast Group Call.

**Initial configuration**

The following pre-conditions are applied for this test case:

1. Three MCS DUTs are connected via a core network using the process as defined in 3GPP TS 33.401.
2. The MCS DUTs are all switched on and have been authorised/authenticated and assigned to a common group.
3. The Requesting DUT and Receiving DUTs are reachable on the Home KMS and Access Network.
4. Functions of the MCPTT server (Originating Participating/Controlling/Terminating Participating MCPTT Function) and all MCPTT DUTs are located in the primary MCPTT system.
5. A pre-arranged group is established with a common MCPTT group ID, GMK and member list on the same Group Management Server (GMS).
6. There are no Emergency or Imminent Peril calls in progress.

**Test Procedure**

**Scenario A: On-demand Broadcast Group Call with Idle floor**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition. A PTT group has been selected by DUT-A | Floor is idle for all DUTs and DUT-A selected the intended group.  **VERIFICATION METHOD:**   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a Broadcast Group Call on their MCPTT device | The following elements are set in the mcptt-info MIME body.  - <mcptt-client-id> shall be identity of DUT-A user  - <broadcast-ind> = true, and  - <mcpttURI>sip:mcptt-group--"UserCurrentlySelectedGroup”  Boadcast Group Call request → MCPTT Server  OPTIONAL: Resolve MCPTT Group X ID→ Group Management Server  MCPTT Server → Broadcast Group Call request to MCPTT DUT-B  MCPTT Server → Broadcast Group Call request to MCPTT DUT-C  Waiting for permission to start a broadcast group call  MCPTT DUT-B → Broadcast Group Call response  MCPTT DUT-C → Broadcast Group Call response  Group management Server grants access to group X and the broadcast group call starts  Media plane and floor control establishment  **VERIFICATION METHOD:**   * Visual verification on GUI of DUT |
| 3 | Check that an audio connection has been established with the PTT group. | The voice connection works in DUT-B and DUT-C  **VERIFICATION METHOD:**   * Audio and visual verification on DUT-B and DUT-C * Audio sent by DUT-A should be audible on DUT-B and DUT-C. MC-PTT User ID of DUT-A user should be visible on DUT-B and DUT-C. |
| 4 | Release the PTT button to release the group call. | MCPTT Server: Generate group call release  Group Call release request to DUT-B and DUT-C (Notify user)  Group Call release response of DUT-B and DUT-C  Releasing floor control and media plane resources associated with group call  The broadcast group call ends  **VERIFICATION METHOD:**   * Audio and visual verification on DUT-B and DUT-C: * No Audio audible anymore on DUT-B and DUT-C. * Active talkburst indication should be cleared on DUT-B and DUT-C. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Broadcast Group Call with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP INVITE for MCPTT broadcast group call | <broadcast-ind> = true in MIME body |
| 2 | <-- | MCPTT Server sends SIP INVITE to DUT-B for MCPTT broadcast group call | <broadcast-ind> = true in MIME body |
| 3 | <-- | MCPTT Server sends SIP INVITE to DUT-C for MCPTT broadcast group call | <broadcast-ind> = true in MIME body |
| 4 | --> | DUT-B sends SIP Ringing and 200 (OK) response to MCPTT Server |  |
| 5 | --> | DUT-C sends SIP Ringing and 200 (OK) response to MCPTT Server |  |
| 6 | <-- | MCPTT Server sends SIP 200 (OK) to DUT-A |  |
| 7 | --> | DUT-A sends audio to MCPTT Server |  |
| 8 | <-- | MCPTT Server sends audio to DUT-B & DUT-C |  |
| Termination of Broadcast Group Call | | | |
| 9 | --> | DUT-A send BYE to server |  |
| 10 | <-- | Server send SIP 200 OK to DUT-A |  |
| 11 | <-- | Server send Disconnect to DUT-B and DUT-C |  |
| 12 | --> | DUT-B send SIP ACK to server |  |
| 13 | --> | DUT-C send SIP ACK to server |  |

**Scenario B: Pre-established session Broadcast Group Call with Idle floor**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition. A PTT group has been selected by DUT-A | Floor is idle for all DUTs and DUT-A selected the intended group.  **VERIFICATION METHOD:**   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a Broadcast Group Call on their MCPTT device | The following elements are set in the mcptt-info MIME body.  - <mcptt-client-id> shall be identity of DUT-A user  - <broadcast-ind> = true, and  - <mcpttURI>sip:mcptt-group--"UserCurrentlySelectedGroup”  Boadcast Group Call request → MCPTT Server  MCPTT Server → Broadcast call response  OPTIONAL: Resolve MCPTT Group X ID→ Group Management Server  MCPTT Server → Broadcast Group Call connect to MCPTT DUT-A  MCPTT Server → Broadcast Group Call connect to MCPTT DUT-B  MCPTT Server → Broadcast Group Call connect to MCPTT DUT-C  Media plane and floor control establishment  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 3 | Check that an audio connection has been established with the MCPTT group. | The voice connection works in DUT-B and DUT-C  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-B and DUT-C, and * Audio sent by DUT-A should be audible on DUT-B and DUT-C. MC-PTT User ID of DUT-A user should be visible on DUT-B and DUT-C. |
| 4 | Release the PTT button to release the group call. | DUT-A releases the floor  Broadcast call disconnect to MCPTT Client B and MCPTT Client C  Releasing floor control and media plane resources associated with group call  The broadcast group call ends  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-B and C: * NO Audio audible anymore on DUT-B and C. * Active talkburst indication should be cleared on DUT-B and C. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Broadcast Group Call with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP REFER for MCPTT broadcast group call | <broadcast-ind> = true in MIME body |
| 2 | <-- | MCPTT Server sends SIP 200K to DUT-B for MCPTT broadcast group call |  |
| 3 | <-- | MCPTT Server sends MCPC CONNECT to DUT-A for MCPTT broadcast group call |  |
| 4 | <-- | MCPTT Server sends MCPC CONNECT to DUT-B and DUT-C for MCPTT broadcast group call | MCPC CONNECT includes GMK |
| 5 | --> | DUT-A sends audio to MCPTT Server |  |
| 6 | <-- | MCPTT Server sends audio to DUT-B & DUT-C |  |
| Termination of Broadcast Group Call | | | |
| 7 | --> | DUT-A sends Floor Release to server |  |
| 8 | <-- | Server send MCPC Disconnect to DUT-B and DUT-C |  |

### 110.2.5 Upgrade of call in progress to emergency or imminent peril call

**Description**

In this test case, an MCPTT DUT will trigger an upgrade to either emergency or imminent peril group call during an in-progress normal group call.

The initial steps are the same as a normal group call but the DUT triggers the emergency or imminent peril upgrade mechanism according to clauses 10.1.1.2.1.3 and 10.1.2.2.1.4 in 3GPP TS 24 379 modelling a scenario of a new elevated risk or incident to the user. In both cases, a re-INVITE is triggered with the new <emergency-ind> or <imminentperil-ind> elements in both on-demand sessions and pre-established sessions.

In this test case all affiliated MCPTT clients are in-progress of a normal group call in an on-demand session or a pre-established session.

This test case will require a minimum of three MCPTT clients which are all on the same MCPTT syetem and Group Management Server (GMS) and are affiliated, and active on the same MCPTT Group.

**Applicability**

3GPP MCS Rel.15 or later

**Related core specifications / References**

3GPP TS 23.379

3GPP TS 24.379

3GPP TS 24.481

**Reason for test**

To verify that the DUT can successfully upgrade an MCPTT normal group call to an emergency group call or to an imminent peril group call in an on-demand session or a pre-established session.

**Initial configuration**

The following pre-conditions are applied for this test case:

1. Three MCS DUTs are connected via a core network using the process as defined in 3GPP TS 33.401.
2. The MCS DUTs are all switched on and have been authorised/authenticated and assigned to a common group.
3. A pre-arranged group is established with a common MCPTT group ID, GMK and member list on the same Group Management Server (GMS)
4. There are no emergency or imminent peril calls in progress in any of the test DUTs.
5. All DUTs are in progress of a group call within a single group.

**Test Procedure**

**Upgrade of in-progress normal group call to an emergency group call or an imminent peril group call**

The test procedure steps are the same for both on-demand sessions and pre-established sessions.

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition. A PTT group has been selected by DUT-A and DUT-A initiates a regular group call to a pre-arranged group in an on-demand session or a pre-established session. | The initial group call is properly established by caller DUT-A to callees DUT-B and DUT-C and is in progress.  The steps can be verified as defined in tests case 1.2.1 Normal Group Call.  **VERIFICATION METHOD:**   * Visual verification on GUI of DUTs |
| 2 | DUT-A upgrades the call to an emergency call or imminent peril call. | a. DUT-A sends a re-INVITE with the following elements set in the mcptt-info MIME body:   * If the upgrade is for emergency group call, an <emergency-ind> element set to "true" * If the upgrade is for imminent peril group call, the <imminentperil-ind> element set to "true"   and Resource-Priority header field populated with the values for an MCPTT emergency group call or for an imminent peril appropriate call as appropriate  b. MCPTT Server sends SIP re-INVITE to MCPTT DUT-B with <emergency-ind> or <imminentperil-ind> set to “true” as appropriate  c. MCPTT Server sends SIP re-INVITE to MCPTT DUT-C with <emergency-ind> or <imminentperil-ind> set to “true” as appropriate  d. Upon receipt of a SIP re-INVITE request the MCPTT client on DUT-B and DUT-C:   * in the case of emegency call upgrade, should display to the MCPTT users the MCPTT ID of the originator of the MCPTT emergency group call and an indication that this is an MCPTT emergency group call * in the case of imminent peril call, should display to the MCPTT users the MCPTT ID of the originator of the MCPTT imminent peril group call and an indication that this is an MCPTT imminent peril group call   e. Emergency call or imminent peril call status is pending in DUT-A  f. MCPTT DUT-B sends SIP 200 OK response accepting emergency or imminent peril group call request  g. MCPTT DUT-C sends SIP 200 OK response accepting emergency or imminent peril group call request  h. MCPC Floor Grant is sent tp DUT-A and MCPC Floor Taken is sent to DUT-B and DUT-C  i. Media plane and floor control are established.  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 3 | Check that an audio connection has been established with the PTT group. | The voice connection works in DUT-B and DUT-C  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-B and DUT-C * Audio sent by DUT-A should be audible on DUT-B and DUT-C. MC-PTT User ID of DUT-A user should be visible on DUT-B and DUT-C. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Upgrade to an emmergency group call or an imminent peril group call from normal group call | | | |
| 1 | -- > | DUT-A sends SIP re-INVITE to upgrade normal group call to an emegency group call or an imminent peril group call | <emergency-ind> = true in MIME body for emergency group call upgrade  <imminentperil-ind> = true in MIME body for imminent peril group call upgrade |
| 2 | <-- | DUT-A received SIP 200 (OK) |  |
| 2 | <-- | MCPTT Server sends SIP re-INVITE to DUT-B for emergency group call or imminent peril group call | <emergency-ind> = true in MIME body for emergency group call upgrade  <imminentperil-ind> = true in MIME body for imminent peril group call upgrade |
| 3 | <-- | MCPTT Server sends SIP re-INVITE to DUT-C emergency group call or imminent peril group calll | <emergency-ind> = true in MIME body for emergency group call upgrade  <imminentperil-ind> = true in MIME body for imminent peril group call upgrade |
| 4 | --> | DUT-B sends SIP 200 (OK) response to MCPTT Server |  |
| 5 | --> | DUT-C sends SIP 200 (OK) response to MCPTT Server |  |
| 6 | <-- | MCPTT Server sends SIP 200 (OK) to DUT-A |  |
| 7 | <-- | DUT-A receives MCPC Floor Grant |  |
| 8 | <-- | DUT-B & DUT-C receive MCPC Floor Taken |  |

### 110.2.6 Floor Control

For the floor control tests we will use specific test configuration for the Devices Under Test (DUT) and the baseline configuration that will be applied across the various test scenarios identified is identified in this section. Here the configuration varies across each test case this will be indicated in each test scenario as appropriate.

#### 110.2.6.1 On-Network Floor Control - Talk/Media Burst - when floor is Idle

**Description**

On-Network, Pre-arranged group establishment for an on-demand Group Call for basic Talk/Media Burst, where the Floor is identified as being idle.

In this test case we have 3 MCX DUTs, of which one is the Initiating DUT that will initiate a floor control request and we will have two Participating DUTs. All three DUTs are active and in the same MCS Group (MCS). The initiating DUT (MCPTT Client A) will initiate a Talk Burst/Media Burst Group call and the Participating DUTs will be notified of the Floor Control progress and session as the group call is set up.

**Applicability**

3GPP MCS Rel.15 or later

**Related core specifications / References**

The following references were used in the development of this test case:

3GPP TS 24.379

3GPP TS 24.380

3GPP TS 24.483

3GPP TS 24.484

**Reason for test**

Verify the MCPTT service is able to establish a successful Floor Control session between multiple participating DUTs in the same Group Call

To verify that a MCPTT Emergency Group Call can be established and provide priority access to the requesting Entity.

**Initial configuration**

The following pre-conditions are applied for this test case:

1. Floor request is initiated when the floor is idle.

2. Multi-Talker Groups are not considered in the test case.

3. No Acknowledgement response messages are required to be sent by the DUTs for the Floor Control messages they receive.

4. Dual Floor Control or Multi-Talker Group will not be applied by the Floor Control Server.

5. Floor Request Queueing is set to “false” on the Floor Control Server.

6. Floor Request Pre-emptive Priority is set to “false”

7. The type of Group Call is identified as a Normal Group Call.

8. The Floor priority for the DUTs is identified as follows:

a. DUT-A: Floor Priority = 10

b. DUT-B: Floor Priority = 10

c. DUT-C: Floor Priority = 10

9. The timers that are configured for each DUT are identified as follows:

a. T100 (Floor Release) = 3S

b. T101 (Floor Request) = 3S

c. T103 (End of RTP media) = 2S

d. T104 (Floor Queue Position Request) = 3S

e. T132 (Queued Request Granted) = 2S

10. The MCS Service Configuration will be common to all DUTs and these parameters are configured as follows:

a. Floor Control Queue – depth = 4

b. Floor Control Queue – max-user-request-time= 30S

c. T1-end-of-rtp-media = 4S

d. T3-stop-talking-grace = 3S

e. T7-floor-idle = 4S

f. T8-floor-revoke = 1S

g. T11-end-of-RTP-dual = 4S

h. T12-stop-talking-dual = 30S

i. T15-conversation = 30S

j. T16-map-group-to-bearer = 0.5S

k. T17-unmap-group-to-bearer = 0.2S

l. T20-floor-granted = 1S

m. T55-connect = 2S

n. T56-disconnect = 2S

o. C7-floor-idle = 10

p. C17-unmap-group-to-bearer = 3

q. C20-floor-granted = 3

r. C55-connect = 3

s. C56-disconnect = 3

**Test procedure**

**Scenario A: Normal Floor Request Process initiated with Idle floor**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | DUT-A initiates a MCPTT floor request session by pressing the PTT Button (HW or SW) on the DUT (and keeping it pressed for the duration of the communication needed) | The MCPTT Floor Control Server identifies that the floor is idle (no ongoing sessions are in progress) & verifies that DUT-A is authorised and permits the request.  A Floor Grant message is sent from the MCPTT Floor Control Server to DUT-A to notify that the floor request has been granted.  The user of DUT-A is informed by a Permission Initiation tone on their device.  **VERIFICATION METHOD:**   * Visual and audio verification on DUT-A |
| 2 | DUT-A takes floor (and is permitted by MCPTT Floor Control Server) | MCPTT Floor Control Server associated with the MCPTT Group (DUT-B & DUT-C) are sent a Floor Taken message.  **VERIFICATION METHOD:**   * log traces on DUT-A, DUT-B and DUT-C |
| 3 | DUT-B tries to take floor after DUT-A has grated floor (and DUT-B is not permitted by MCPTT Floor Control Server) | DUT-B receives Floor Taken message by MCPTT Floor Control Server  **VERIFICATION METHOD:**   * Visual verification on DUT-B (GUI) or log files |
| 4 | DUT-C tries to take floor after DUT-A has grated floor (and DUT-C is not permitted by MCPTT Floor Control Server) | DUT-C receives Floor Taken message by MCPTT Floor Control Server  **VERIFICATION METHOD:**   * Visual verification on DUT-C (GUI) or log files |
| 5 | DUT-A starts sending audio | Upon successful receipt of the Floor Grant message, DUT-A will start to forward RTP (voice) media packets toward the Controlling MCPTT Server  **VERIFICATION METHOD:**   * Log traces on DUT-A |
| 6 | Controlling MCPTT Server passing on audio to DUT-B and DUT-C | The Controlling MCPTT Server distributes the RTP (voice) media to all the other MCPTT DUT in the MCPTT group call.  **VERIFICATION METHOD**:   * Log traces on on MCPTT AS or DUT-B / DUT-CAudio and visual verification on DUT-B, DUT-C:Squelch should be indicated and audio from DUT-A should be audibleVisual and audio verification on DUT-B and DUT-C |
| 7 | DUT-A releases PTT | When DUT-A has completed their communication, they will release the PTT Button - this action will send a Floor Release message from the DUT-A towards the MCPTT Floor Control Server.  This action will also result in the termination of the RTP (voice) Media at the same time  **VERIFICATION METHOD:**   * Log traces on server side, or * Log traces on DUT-A DUT-A stops sending audio to MCPTT AS (log traces on DUT) |
| 8 | floor idle (Verfied / indicated by Floor Control Server) | Upon receiving the Floor Release message from DUT-A, the MCPTT Floor Control Server will send a Floor Idle message to all the DUTs in the MCPTT group call.  With the release of the RTP Media by DUT-A, this will also result in the RTP (voice) media session to be terminated for all DUTs in the MCPTT Group call.  **VERIFICATION METHOD**:   * Audio (no Audio) and visual verification on DUT-B and DUT-C.   + Squelch indication should be cleared.(Floor IDLE) (log tracing) |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Emergency Group Call with Idle floor | | | |
| 1 | -- > | DUT-A initiates a Floor Request |  |
| 2 | <-- | MCPTT Floor Control Server responds with Floor Granted |  |
| 3 | <-- | MCPTT Floor Control Server notifies to DUT-B that Floor is Taken |  |
| 4 | <-- | MCPTT Floor Control Server notifies to DUT-C that Floor is Taken |  |
| 5 | --> | DUT-A transmits RTP Voice Media |  |
| 6 | <-- | MCPTT Server sends audio to DUT-B & DUT-C |  |
| 7 | --> | DUT-A releases Floor and Voice Media |  |
| 8 | <-- | Floor & Voice Media is released towards DUT-B & DUT-C |  |
| 9 | <-- | Floor Idle notification sent to DUT-A, DUT-B & DUT-C. |  |

**Details flow sequence diagram:**

A picture containing text, screenshot, diagram, parallel

Description automatically generated

#### 110.2.6.2 On-Network Floor Control requested whilst floor is taken, and Queuing is not applied

**Description**

On-Network, Pre-arranged group establishment for an on-demand Group Call for basic Talk/Media Burst, where the Floor Control Server does not offer Floor Queuing and a Floor Request is initiated whilst an ongoing Media Bust is in progress.

In this test case we have 3 MCX DUTs, of which one is the Initiating DUT that will initiate a floor control request and we will have two Participating DUTs. All three DUTs are active and in the same MCS Group (MCS). The initiating DUT (MCPTT Client A) will initiate a Talk Burst/Media Burst Group call and the Participating DUTs will be notified of the Floor Control progress and session as the group call is set up.

**Applicability**

3GPP MCS Rel.15 or later

**Related core specifications / References**

The following references were used in the development of this test case:

3GPP TS 24.379

3GPP TS 24.380

3GPP TS 24.483

3GPP TS 24.484

**Reason for test**

Verify the MCPTT service is able to establish a successful Floor Control session between multiple participating DUTs in the same Group Call

To verify that a MCPTT Emergency Group Call can be established and provide priority access to the requesting Entity.

**Initial configuration**

Same as in Test Case 110.2.6.1

Note: for this test scenario all DUTs have equal Floor Priority of 10.

**Test procedure**

**Scenario B: Floor Request Initiated whilst a Floor Grant is in progress where DUT have same Floor Priority**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | DUT-A initiates a MCPTT floor request session by pressing the PTT Button (HW or SW) on the DUT (and keeping it pressed for the duration of the communication needed) | The MCPTT Floor Control Server identifies that the floor is idle (no ongoing sessions are in progress) & verifies that DUT-A is authorised and permits the request.  The following State Changes are applied:DUT1 State Change: [U: Has No Permission] → [U: Pending Request]  A Floor Grant message is sent from the MCPTT Floor Control Server to DUT-A to notify that the floor request has been granted.  The user of DUT-A is informed by a Permission Initiation tone on their device.  The following State Changes are applied:   * Floor Control Server state change [G: Floor Idle] à [G: Floor Taken] * DUT-A State Change: [U: Pending Request] à [U: Has Permission]   **VERIFICATION METHOD:**   * Visual verification on DUT-A,  1. PTT indicated as Requested, log traces: |
| 2 | DUT-A takes floor (and is permitted by MCPTT Floor Control Server) | MCPTT Floor Control Server associated with the MCPTT Group (DUT-B & DUT-C) are sent a Floor Taken message.  **VERIFICATION METHOD**:   * log traces on DUT-A, DUT-B and DUT-C |
| 3 | DUT-A starts sending audio | Upon successful receipt of the Floor Grant message, DUT-A will start to forward RTP (voice) media packets toward the Controlling MCPTT Server  **VERIFICATION METHOD:**   * Log traces on DUT-A |
| 4 | Controlling MCPTT Server passing on audio to DUT-B and DUT-C | The Controlling MCPTT Server distributes the RTP (voice) media to all the other MCPTT DUT in the MCPTT group call.  **VERIFICATION METHOD:**   * Log traces on on MCPTT AS or DUT-B / DUT-C  1. Audio and visual verification on DUT-B, DUT-C: 2. Squelch should be indicated and audio from DUT-A should be audibleVisual and audio verification on DUT-B and DUT-C |
| 5 | DUT-A releases PTT | When DUT-A has completed their communication, they will release the PTT Button - this action will send a Floor Release message from the DUT-A towards the MCPTT Floor Control Server.  This action will also result in the termination of the RTP (voice) Media at the same time  **VERIFICATION METHOD:**   * Log traces on server side, or * Log traces on DUT-A, where  1. DUT-A stops sending audio to MCPTT PAS (log traces on DUT) |
| 6 | floor idle (Verfied / indicated by Floor Control Server) | Upon receiving the Floor Release message from DUT-A, the MCPTT Floor Control Server will send a Floor Idle message to all the DUTs in the MCPTT group call.  With the release of the RTP Media by DUT-A, this will also result in the RTP (voice) media session to be terminated for all DUTs in the MCPTT Group call.  **VERIFICATION METHOD**:   * Audio (no Audio) and visual verification on DUT-B and DUT-C. Squelch indication should be cleared.(Floor IDLE) (log tracing) |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Emergency Group Call with Idle floor | | | |
| 1 | -- > | DUT-A initiates a Floor Request |  |
| 2 | <-- | MCPTT Floor Control Server responds with Floor Granted |  |
| 3 | <-- | MCPTT Floor Control Server notifies to DUT-B that Floor is Taken |  |
| 4 | <-- | MCPTT Floor Control Server notifies to DUT-C that Floor is Taken |  |
| 5 | --> | DUT-A transmits RTP Voice Media |  |
| 6 | <-- | MCPTT Server sends audio to DUT-B & DUT-C |  |
| 7 | --> | DUT-B initiates a Floor Request |  |
| 8 | <-- | MCPTT Server sends Deny Floor notification to DUT-B |  |

**Details flow sequence diagram:**

The following shows the sequence flow for this test case.

**A screenshot of a computer program

Description automatically generated with low confidence**

### 110.2.7 Priorities

**Description**

The purpose of this test is to validate the successful user authentication including a successful MCX (using PTT placeholder for DATA, Video) Client registration:

**Applicability**

3GPP MCS Rel. 15 or later

**Related core specifications**

3GPP TS 24 380

3GPP TS 24.481

3GPP TS 33.102, 5.1.2

3GPP TS 33.401, 6.1

3GPP TS 33.179

3GPP TS 33.180

**Reason for test**

on-network vs. off-network floor control? Difference in priorities

Does it make sense to have an own chapter on priorities?

Testing effect of priorities on media burst (in group call):

* Preemptiveness at floor control
* Token / Queueing
* Late entry

**Initial configuration**

Components:

* DUT-A: mobile device with Client App
* DUT-B:
* DUT-C:
* IdMS
* MS core, Network and Cell
* DUT has a MCX capable application
* DUT is switched off
* DUT has valid SIM for test network
* Minimum is that connectivity should be established between all elements of the specific scenario (and tested prior this case)
* The IdMS shall have the MC User, Service and Credential pre-provisioned and authentication must be performed over the CSC-1 reference point (between the IdM Client and IdM Server)
* (The CSC-1 shall consist of two endpoints; the authorization endpoint and token endpoint.)
* Authentication is performed using Open Id Connect 1.0

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | Power on DUT-And confirm successful attach procedure. | DUT registers – after contacting the network to get the subscriber data and authentication data – successfully on Cell of a (home) network. |
| 2 | Open MCX Client Application. | Authentication Request will appear  Network service is indicated on the display on DUT with MCX Client Application |
| 3 | Log into the MCX Application using a valid account (Username and Password). | MCX User successfully authenticates (and is authorized to use its MCX services)   * that includes token is requested by MCX client * User is verified and data and identity token are correctly received according 3GPP TS 24 482 * Following Service Authorisation test case: functions available with the user (rights) are appearing  (to be defined)   Background activities coming with this test on IMS Core: MCPTT Client Registers & S-CSCF send 3rd party registration, after MCPTT User has already authenticated with IdMS |

**Example message flow for MCX User authentication to IdMS:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| User Registration | | | |
| 1 | --> | MCX User send authorisation request to IdMS |  |
| 2 | <-- | IdMS redirects Authorisation mechanism |  |
| 3 | --> | MCX User sends Token Request |  |
| 4 | --> | IdMS sends Token Response |  |

## 110.3 Group Call, chat, on-demand and pre-established session

### 110.3.1 Chat Group Call

**Description**

In a chat group (restricted) call model, the MCPTT user individually joins a group call without being invited. The establishement of a chat group (restricted) call does not result in other group members being invited. “Restricted” means that only users that have been configured as members of the given group are allowed to join the group communications for the given group.

The DUT (with MCX client application) shall successfully perform a “Chat Group Call” with an MCX server.

**Applicability**

3GPP Rel. 15 or later

**Related core specifications**

3GPP TS 33.180

3GPP TS 23.379

3GPP TS 24.379

**Reason for test**

To verify the DUT-Can successfully establish a Chat Group Call between different users.

**Initial configuration**

The MCS service provider supports Chat Group Call.

The MCPTT clinets of DUT-A, DUT-B and DUT-C have joined (affiliated to) a common group.

All devices are registered and have successfully performed user authentication and been assigned a valid token.

The DUTs are authorised to use MCPTT service.

There are no group normal, emergency or imminent peril calls in progress.

All three DUTs may have activated functional alias(es) configured to be used during the group call communication.

**Test procedure**

**Scenario A: On-demand Chat Group Call**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | DUT-A indicates to join the group communication for the group. | DUT-A sends a group join request with the |MCPTT group ID of the group, its media parameter and optionally a functional alias and/or an implicit floor request.  The MCPTT server replies with a group join response indicating the acceptance of the group join request and also returns the MCPTT server selected media parameters for the group call in group call response.  DUT-B and DUT-C do not receive any indication of a group call arrival  **VERIFICATION METHOD:**   * Visual verification of GUI of DUT-A and PCAP traces for media parameters. |
| 2 | DUT-B indicates to join the group communication for the group | DUT-B sends a group join request with the |MCPTT group ID of the group, its media parameter and optionally a functional alias and/or an implicit floor request.  The MCPTT server replies with a group join response indicating the acceptance of the group join request and also returns the MCPTT server selected media parameters for the group call in group call response.  DUT-A and DUT-C do not receive any indication of a group call arrival  **VERIFICATION METHOD:**   * Visual verification of GUI of DUT-B and PCAP traces for media parameters. |
| 3 | DUT-C indicates to join the group communication for the group | DUT-C sends a group join request with the |MCPTT group ID of the group, its media parameter and optionally a functional alias and/or an implicit floor request.  The MCPTT server replies with a group join response indicating the acceptance of the group join request and also returns the MCPTT server selected media parameters for the group call in group call response.  DUT-A and DUT-B do not receive any indication of a group call arrival  **VERIFICATION METHOD:**   * Visual verification of GUI of DUT-B and PCAP traces for media parameters. |
| 4 | DUT-A requests the floor to transmit | DUT-A requests the floor  MCPTT server establishes the media plane for the call.  DUT-A receives the floor grant.  DUT-B and DUT-C receives the MCPTT floor taken.  OPTIONAL: If present, the functional alias of DUT-A, DUT-B and DUT-C are displayed as appropriate.  Media is transmitted by DUT-A and received by DUT-B and DUT-C  **VERIFICATION METHOD:**   * Visual verification on GUI of DUTs and hearing of audio at DUT-B and DUT-C. |
| 5 | Media connection is established between DUT-A, DUT-B and DUT-C  Floor control is used by the floor participants associated with DUT-A, DUT-B and DUT-C to take and release the floor among them. | The voice connection works to B and C  **VERIFICATION METHOD:**   * Audio and visual verification on DUT-B and C * Audio sent by DUT-A should be audible on DUT-B and C. MC-PTT User ID of DUT-A user should be visible on DUT-B and C. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Chat Group Call | | | |
| 1 | --> | DUT-A sends chat group SIP INVITE to the server | < session\_type> chat< / session\_type>  <mcpt t - request- uri Type= "Normal"> <mcpttURI> sip:mcpt t - chat- group- A@example.com< /mcpttURI> < / mcpt t - request- uri> |
| 2 | <-- | MCPTT server sends SIP 200 OK to DUT-A |  |
| 3 | --> | DUT-B sends chat group SIP INVITE to the server | < session\_type> chat< / session\_type>  <mcpt t - request- uri Type= "Normal"> <mcpttURI> sip:mcpt t - chat- group- A@example.com< /mcpttURI> < / mcpt t - request- uri> |
| 4 | <-- | MCPTT server sends SIP 200 OK to DUT-B |  |
| 5 |  | DUT-C sends chat group SIP INVITE to the server | < session\_type> chat< / session\_type>  <mcpt t - request- uri Type= "Normal"> <mcpttURI> sip:mcpt t - chat- group- A@example.com< /mcpttURI> < / mcpt t - request- uri> |
| 6 | <-- | MCPTT server sends SIP 200 OK to DUT-C |  |
| 7 | --> | DUT-B sends floor request |  |
| 8 | <-- | MCPTT server sends floor granted to DUT-B and floor taken to DUT-A and DUT-C |  |
| 9 | <--> | Audio flows between DUT-B, DUT-A and DUT-C |  |

**Scenario B: Pre-established session Chat Group Call**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | DUT-A has already setup a pre-established session and indicates to join the group communication for the group. | DUT-A sends a group join request with the MCPTT group ID of the group, its media parameter and optionally a functional alias and/or an implicit floor request.  The MCPTT server replies with a group join response indicating the acceptance of the group join request and also returns the MCPTT server selected media parameters for the group call in group call response.  DUT-B and DUT-C do not receive any indication of a group call arrival  **VERIFICATION METHOD:**   * Visual verification of GUI of DUT-A and PCAP traces for media parameters. |
| 2 | DUT-B has already setup a pre-established session and indicates to join the group communication for the group | DUT-B sends a group join request with the |MCPTT group ID of the group, its media parameter and optionally a functional alias and/or an implicit floor request.  The MCPTT server replies with a group join response indicating the acceptance of the group join request and also returns the MCPTT server selected media parameters for the group call in group call response.  DUT-A and DUT-C do not receive any indication of a group call arrival  **VERIFICATION METHOD:**   * Visual verification of GUI of DUT-B and PCAP traces for media parameters. |
| 3 | DUT-C has already setup a pre-established session and indicates to join the group communication for the group | DUT-C sends a group join request with the |MCPTT group ID of the group, its media parameter and optionally a functional alias and/or an implicit floor request.  The MCPTT server replies with a group join response indicating the acceptance of the group join request and also returns the MCPTT server selected media parameters for the group call in group call response.  DUT-A and DUT-B do not receive any indication of a group call arrival  **VERIFICATION METHOD:**   * Visual verification of GUI of DUT-B and PCAP traces for media parameters. |
| 4 | DUT-A requests the floor to transmit | DUT-A requests the floor  MCPTT server establishes the media plane for the call.  DUT-A receives the floor grant.  DUT-B and DUT-C receives the MCPTT floor taken.  OPTIONAL: If present, the functional alias of DUT-A, DUT-B and DUT-C are displayed as appropriate.  Media is transmitted by DUT-A and received by DUT-B and DUT-C **VERIFICATION METHOD:**   * Visual verification on GUI of DUTs and hearing of audio at DUT-B and DUT-C. |
| 5 | Media connection is established between DUT-A, DUT-B and DUT-C  Floor control is used by the floor participants associated with DUT-A, DUT-B and DUT-C to take and release the floor among them. | The voice connection works to B and C  **VERIFICATION METHOD:**   * Audio and visual verification on DUT-B and C * Audio sent by DUT-A should be audible on DUT-B and C. MC-PTT User ID of DUT-A user should be visible on DUT-B and C. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Chat Group Call | | | |
| 1 | --> | DUT-A sends chat group SIP REFER to the server | the application/resource-lists MIME body a single <entry> element containing a "uri" attribute set to the chat group identity  the application/vnd.3gpp.mcptt-info MIME body with the <session-type> element set to a value of "chat"; and the <mcptt-client-id> element set to the MCPTT client ID of the originating MCPTT client |
| 2 | <-- | MCPTT server sends SIP 200 OK to DUT-A |  |
| 3 | --> | DUT-B sends chat group SIP REFER to the server | the application/resource-lists MIME body a single <entry> element containing a "uri" attribute set to the chat group identity  the application/vnd.3gpp.mcptt-info MIME body with the <session-type> element set to a value of "chat"; and the <mcptt-client-id> element set to the MCPTT client ID of the originating MCPTT client |
| 4 | <-- | MCPTT server sends SIP 200 OK to DUT-B |  |
| 5 |  | DUT-C sends chat group SIP REFER to the server | the application/resource-lists MIME body a single <entry> element containing a "uri" attribute set to the chat group identity  the application/vnd.3gpp.mcptt-info MIME body with the <session-type> element set to a value of "chat"; and the <mcptt-client-id> element set to the MCPTT client ID of the originating MCPTT client |
| 6 | <-- | MCPTT server sends SIP 200 OK to DUT-C |  |
| 7 | --> | DUT-B sends floor request |  |
| 8 | <-- | MCPTT server sends floor granted to DUT-B and floor taken to DUT-A and DUT-C |  |
| 9 | <--> | Audio flows between DUT-B, DUT-A and DUT-C |  |

## 110.4 Private Call, on-demand and pre-established session

### 110.4.1 Automatic commencement mode

**Description**

In this test case, an MCPTT DUT will trigger a private call to the other MCPTT DUT in an automatic commencement mode. Automatic commencement mode indicates the terminating client will take the call without interacting with the user holding the terminating DUT.

There are four test cases of a private call in automatic commnencement mode, namely:

1. On-demand private call with floor control (half-duplex mode)
2. On-demand private call without floor control (full duplex mode)
3. Pre-established session private call with floor control (half duplex mode)
4. Pre-establhsed session private call without floor control (full duplex mode)

In this scenario a user, MCPTT DUT-A, decides to establish a private call (not existing one yet). The call is made using automatic commencement mode.

In this test case all affiliated MCPTT clients are in idle mode.

This test case will require a minimum of two DUTs with MCPTT clients which are registered on the same MCPTT server.

**Applicability**

3GPP MCS Rel.15 or later

**Related core specifications / References**

3GPP TS 23.379

3GPP TS 24.379

3GPP TS 24.380

**Reason for test**

Verify that a private call can be initiated in automatic commencement mode.

**Initial configuration**

The following pre-conditions are applied for this test case:

1. Two MCS DUTs (DUT-A and DUT-B) are connected via a core network using the process as defined in 3GPP TS 33.401.
2. The MCS DUTs are all switched on and have been authorised/authenticated.
3. The Originating DUT and Receiving DUT are reachable on the Home KMS and Access Network.
4. Functions of the MCPTT server (Originating Participating/Controlling/Terminating Participating MCPTT Function) and all MCPTT DUTs are located in the primary MCPTT system.
5. There are no Emergency or Imminent Peril calls in progress.

**Test Procedure**

**Scenario A: On-demand private call on automatic commencement mode with floor control**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition: DUT-A and DUT-B are on idle call mode and no pre-established sessions are on | Floor is idle for all DUTs.  **VERIFICATION METHOD:**   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a private call on-demand to DUT-B | DUT-A sends “Private Call Setup Request” with the commencement mode set to “Automatic Commencement Mode” and with floor control (half duplex mode).  **VERIFICATION METHOD:**   * Visual verification on GUI of DUT |
| 3 | DUT-B receives “Private Call Setup Request” | DUT-B receives the caller ID with other identification parameters.  DUT-B sends “Private Call Accept” and establishes a media session.  **VERIFICATION METHOD:**   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-A should be audible on DUT-B. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| On-demand private call on automatic commencement mode with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP INVITE for on-demand private call | Answer Mode = Auto and SDP offer **contains** one SDP "m=application" media-level section for a media-floor control entity |
| 2 | < -- | MCPTT SIP Core sends SIP 100 Trying to DUT-A |  |
| 3 | < -- | MCPTT SIP Core sends SIP INVITE to DUT-B for on-demand private call | Answer Mode = Auto and SDP offer **contains** one SDP "m=application" media-level section for a media-floor control entity |
| 4 | --> | DUT-B sends SIP 100 Trying to MCPTT SIP Core |  |
| 4 | --> | DUT-B sends SIP 200 (OK) to MCPTT SIP Core |  |
| 5 | < -- | MCPTT SIP Core sends SIP 200 (OK) DUT-A |  |
| 6 | --> | DUT-A sends SIP ACK to MCPTT SIP Core |  |
| 7 | < -- | MCPTT server sends “Floor Taken by DUT-A” to DUT-B |  |
| 8 | <-- > | DUT-A sends audio to DUT-B |  |

**Scenario B: On-demand private call on automatic commencement mode without floor control**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition: DUT-A and DUT-B are on idle call mode | Floor is idle for all DUTs.  **VERIFICATION METHOD:**   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a private call on-demand to DUT-B | DUT-A sends “Private Call Setup Request” with the commencement mode set to “Automatic Commencement Mode” without floor control (full duplex mode).  **VERIFICATION METHOD:**   * Visual verification on GUI of DUT |
| 3 | DUT-B receives “Private Call Setup Request” | DUT-B receives the caller ID with other identification parameters.  DUT-B sends “Private Call Accept” and establishes a media session.  **VERIFICATION METHOD:**   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-A should be audible on DUT-B. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| On-demand private call on automatic commencement mode with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP INVITE for on-demand private call | Answer Mode = Auto and and SDP offer **does not contain** one SDP "m=application" media-level section for a media-floor control entity |
| 2 | <-- | MCPTT SIP Core sends SIP 100 Trying to DUT-A |  |
| 3 | <-- | MCPTT SIP Core sends SIP INVITE to DUT-B for on-demand private call | Answer Mode = Auto and and and SDP offer **does not contain** one SDP "m=application" media-level section for a media-floor control entity |
| 4 | --> | DUT-B sends SIP 100 Trying to MCPTT SIP Core |  |
| 5 | --> | DUT-B sends SIP 200 (OK) to MCPTT SIP Core |  |
| 6 | <-- | MCPTT SIP Core sends SIP 200 (OK) DUT-A |  |
| 7 | <--> | DUT-A sends audio to DUT-B and vice versa |  |

**Scenario C: Pre-established session private call on automatic commencement mode with floor control**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition: DUT-A and DUT-B are on idle call mode but have done SIP registration and preestablished session establishment with **no** implicit floor control request | Floor is idle for all DUTs.  **VERIFICATION METHOD:**   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a private call to DUT-B | DUT-A sends “Private Call Setup Request” with the commencement mode set to “Automatic Commencement Mode” with floor control (half duplex mode).  **VERIFICATION METHOD:**   * Visual verification on GUI of DUT |
| 3 | DUT-B receives “Private Call Setup Request” | DUT-B receives the caller ID with other identification parameters.  DUT-B sends “Private Call Accept” and establishes a media session without user interaction  **VERIFICATION METHOD:**   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-A should be audible on DUT-B. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Pre-established session private call on automatic commencement mode with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP REFER to MCPTT SIP Core for MCPTT private call | Answer Mode = auto and SDP offer **contains** one SDP "m=application" media-level section for a media-floor control entity |
| 2 | <-- | MCPTT SIP Core sends SIP re-INVITE to DUT-B for MCPTT private call | Answer Mode = auto and SDP offer **contains** one SDP "m=application" media-level section for a media-floor control entity |
| 3 | -- > | DUT-B sends SIP 200 (OK) to MCPTT SIP Core |  |
| 4 | <-- | MCPTT SIP Core sends SIP 200 (OK) to DUT-A |  |
| 5 | <-- | MCPTT Server sends MCPC CONNECT to DUT-A |  |
| 6 | <-- | MCPTT Server sends MCPC CONNECT to DUT-B | Inviting User ID of DUT-A is included |
| 7 | -- > | DUT-A sends MCPC ACK to MCPTT Server |  |
| 8 | --> | DUT-B sends MCPC ACK to MCPTT Sever |  |
| 9 | <-- | MCPTT server sends “Floor Taken by DUT-A” to DUT-B |  |
| 10 | <-- > | DUT-A sends audio to DUT-B and vice versa |  |

**Scenario D: Pre-established session private call on automatic commencement mode without floor control**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition: DUT-A and DUT-B are on idle call mode but have done SIP registration and preestablished session establishment with **no** implicit floor control | Floor is idle for all DUTs.  **VERIFICATION METHOD:**   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a private call to DUT-B | DUT-A sends “Private Call Setup Request” with the commencement mode set to “Automatic Commencement Mode” without floor control (full duplex mode).  **VERIFICATION METHOD:**   * Visual verification on GUI of DUT |
| 3 | DUT-B receives “Private Call Setup Request” | DUT-B receives the caller ID with other identification parameters.  DUT-B sends “Private Call Accept” and establishes a media session without user interaction  **VERIFICATION METHOD:**   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-A should be audible on DUT-B. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Pre-established session private call on automatic commencement mode with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP REFER to MCPTT SIP Core for MCPTT private call | Answer Mode = Auto and and SDP offer **does not contain** one SDP "m=application" media-level section for a media-floor control entity |
| 2 | <-- | MCPTT SIP Core sends SIP re-INVITE to DUT-B for MCPTT private call | Answer Mode = Auto and and SDP offer **does not contain** one SDP "m=application" media-level section for a media-floor control entity |
| 3 | -- > | DUT-B sends SIP 200 (OK) to MCPTT SIP Core |  |
| 4 | <-- | MCPTT SIP Core sends SIP 200 (OK) to DUT-A |  |
| 5 | <-- | MCPTT Server sends MCPC CONNECT to DUT-A |  |
| 6 | <-- | MCPTT Server sends MCPC CONNECT to DUT-B | Inviting User ID of DUT-A is included |
| 7 | -- > | DUT-A sends MCPC ACK to MCPTT Server |  |
| 8 | --> | DUT-B sends MCPC ACK to MCPTT Sever |  |
| 9 | <-- > | DUT-A and DUT-B exchange audio vice versa |  |

### 110.4.2 Manual commencement mode

**Description**

In this test case, an MCPTT DUT will trigger a private call to the other MCPTT DUT in a manual commencement mode. Manual commencement mode indicates the terminating client will present the call to the receiving MCPTT user for their interaction with the client to accept or reject the incoming private call setup.

There are four test cases of a private call on manual commnencement mode, namely:

1. On-demand private call with floor control (half-duplex mode)
2. On-demand private call without floor control (full duplex mode)
3. Pre-established session private call with floor control (half duplex mode)
4. Pre-establhsed session private call without floor control (full duplex mode)

In this scenario a user, MCPTT DUT-A, decides to establish a private call (not existing one yet). The call is made using manual commencement mode.

In this test case all affiliated MCPTT clients are in idle mode.

This test case will require a minimum of two DUTs with MCPTT clients which are registered on the same MCPTT server.

**Applicability**

3GPP MCS Rel.15 or later

**Related core specifications / References**

3GPP TS 23.379

3GPP TS 24.379

3GPP TS 24.380

**Reason for test**

Verify that a private call can be initiated on manual commencement mode.

**Initial configuration**

The following pre-conditions are applied for these test cases:

1. Two MCS DUTs (DUT-A and DUT-B) are connected via a core network using the process as defined in 3GPP TS 33.401.
2. The MCS DUTs are all switched on and have been authorised/authenticated.
3. The Originating DUT and Receiving DUT are reachable on the Home KMS and Access Network.
4. Functions of the MCPTT server (Originating Participating/Controlling/Terminating Participating MCPTT Function) and all MCPTT DUTs are located in the primary MCPTT system.
5. There are no Emergency or Imminent Peril calls in progress.

**Test Procedure**

**Scenario A: On-demand private call on manual commencement mode with floor control**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition: DUT-A and DUT-B are on idle call mode and no pre-established sessions are on | Floor is idle for all DUTs.  **VERIFICATION METHOD**:   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a private call on-demand to DUT-B | DUT-A sends “Private Call Setup Request” with the commencement mode set to “Manual Commencement Mode” and with floor control (half duplex mode).  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 3 | DUT-B receives “Private Call Setup Request” with a ringing indication and accepts the call | DUT-B receives a “Private Call Ringing” with the caller ID and other identification parameters.  DUT-B user manually accepts the incoming call.  DUT-B sends “Private Call Accept” and establishes a media session.  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-A should be audible on DUT-B. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| On-demand private call on manual commencement mode with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP INVITE for on-demand private call | Answer Mode = Manual and SDP offer **contains** one SDP "m=application" media-level section for a media-floor control entity |
| 2 | < -- | MCPTT SIP Core sends SIP 100 Trying to DUT-A |  |
| 3 | < -- | MCPTT SIP Core sends SIP INVITE to DUT-B for on-demand private call | Answer Mode = Manual and SDP offer **contains** one SDP "m=application" media-level section for a media-floor control entity |
| 4 | --> | DUT-B sends SIP 100 Trying to MCPTT SIP Core |  |
| 5 | --> | DUT-B sends SIP 180 Ringing to MCPTT SIP Core |  |
| 6 | --> | DUT-B sends SIP 200 (OK) to MCPTT SIP Core | Only after the recipient has manually accepted the incoming call |
| 7 | < -- | MCPTT SIP Core sends SIP ACK to DUT-B |  |
| 8 | < -- | MCPTT SIP Core sends SIP 200 (OK) DUT-A |  |
| 9 | --> | DUT-A sends SIP ACK to MCPTT SIP Core |  |
| 10 | < -- | MCPTT server sends “Floor Taken by DUT-A” to DUT-B |  |
| 11 | <-- > | DUT-A sends audio to DUT-B |  |

**Scenario B: On-demand private call on manual commencement mode without floor control**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition: DUT-A and DUT-B are on idle call mode and and no pre-established sessions are on | Floor is idle for all DUTs.  **VERIFICATION METHOD**:   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a private call on-demand to DUT-B | DUT-A sends “Private Call Setup Request” with the commencement mode set to “Manual Commencement Mode” without floor control (full duplex mode).  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 3 | DUT-B receives “Private Call Setup Request” with a ringing indication and accepts the call | DUT-B receives a “Private Call Ringing” with the caller ID and other identification parameters..  DUT-B sends “Private Call Accept” and establishes a media session.  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-A should be audible on DUT-B. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| On-demand private call on manual commencement mode with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP INVITE for on-demand private call | Answer Mode = Manual and and SDP offer **does not contain** SDP "m=application" media-level section for a media-floor control entity |
| 2 | <-- | MCPTT SIP Core sends SIP 100 Trying to DUT-A |  |
| 3 | <-- | MCPTT SIP Core sends SIP INVITE to DUT-B for on-demand private call | Answer Mode = Manual and and and SDP offer **does not contain** SDP "m=application" media-level section for a media-floor control entity |
| 4 | --> | DUT-B sends SIP 100 Trying to MCPTT SIP Core |  |
| 5 | --> | DUT-B sends SIP 180 Ringing to MCPTT SIP Core |  |
| 6 | --> | DUT-B sends SIP 200 (OK) to MCPTT SIP Core | Only after the recipient has manually accepted the incoming call |
| 7 | <-- | MCPTT SIP Core sends SIP ACK to DUT-B |  |
| 8 | <-- | MCPTT SIP Core sends SIP 200 (OK) DUT-A |  |
| 9 | --> | DUT-A sends SIP ACK to MCPTT SIP Core |  |
| 10 | <--> | DUT-A sends audio to DUT-B and vice versa |  |

**Scenario C: Pre-established session private call on manual commencement mode with floor control**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition: DUT-A and DUT-B are on idle call mode but have done SIP registration and preestablished session establishment with **no** implicit floor control request | Floor is idle for all DUTs.  **VERIFICATION METHOD**:   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a private call to DUT-B | DUT-A sends “Private Call Setup Request” with the commencement mode set to “Manual Commencement Mode” with floor control (half duplex mode).  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 3 | DUT-B receives “Private Call Setup Request” with a ringing indication and the user accepts the call | DUT-B receives a “Private Call Ringing” with the caller ID with other identification parameters.  DUT-B user manually accepts the incoming call.  DUT-B sends “Private Call Accept” and establishes a media session after the user manually accepts the call.  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-A should be audible on DUT-B. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Pre-established session private call on manual commencement mode with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP REFER to MCPTT SIP Core for MCPTT private call | Answer Mode = manual and SDP offer **contains** one SDP "m=application" media-level section for a media-floor control entity |
| 2 | <-- | MCPTT SIP Core sends SIP re-INVITE to DUT-B for MCPTT private call | Answer Mode = manual and SDP offer **contains** one SDP "m=application" media-level section for a media-floor control entity |
| 3 | <-- | MCPTT SIP Core sends SIP 200 (OK) to DUT-A |  |
| 4 | -- > | DUT-B sends SIP 180 Ringing to MCPTT SIP Core |  |
| 5 | -- > | DUT-B sends SIP 200 (OK) to MCPTT SIP Core | Only after the recipient has manually accepted the incoming call |
| 6 | <-- | MCPTT SIP Core sends SIP ACK to DUT-B |  |
| 7 | <-- | MCPTT Server sends MCPC CONNECT to DUT-A |  |
| 8 | <-- | MCPTT Server sends MCPC CONNECT to DUT-B | Inviting User ID of DUT-A is included |
| 9 | -- > | DUT-A sends MCPC ACK to MCPTT Server |  |
| 10 | --> | DUT-B sends MCPC ACK to MCPTT Sever |  |
| 11 | <-- | MCPTT server sends “Floor Taken by DUT-A” to DUT-B |  |
| 12 | <-- > | DUT-A sends audio to DUT-B and vice versa |  |

**Scenario D: Pre-established session private call on manual commencement mode without floor control**

| **-** | **Test procedure** | **Expected behaviour** |
| --- | --- | --- |
| 1 | Start condition: DUT-A and DUT-B are on idle call mode but have done SIP registration and preestablished session establishment with **no** implicit floor control | Floor is idle for all DUTs.  **VERIFICATION METHOD**:   * Visual verification on GUI of DUTs |
| 2 | DUT-A initiates a private call to DUT-B | DUT-A sends “Private Call Setup Request” with the commencement mode set to “Manual Commencement Mode” without floor control (full duplex mode).  **VERIFICATION METHOD**:   * Visual verification on GUI of DUT |
| 3 | DUT-B receives “Private Call Setup Request” with a ringing indication and the user accepts the call | DUT-B receives a “Private Call Ringing” with the caller ID and other identification parameters.  DUT-B user manually accepts the incoming call.  DUT-B sends “Private Call Accept” and establishes a media session after the user manually accepts the call.  **VERIFICATION METHOD**:   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-A should be audible on DUT-B. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Pre-established session private call on manual commencement mode with Idle floor | | | |
| 1 | -- > | DUT-A sends SIP REFER to MCPTT SIP Core for MCPTT private call | Answer Mode = Manual and and SDP offer **does not contain** any SDP "m=application" media-level section for a media-floor control entity |
| 2 | <-- | MCPTT SIP Core sends SIP re-INVITE to DUT-B for MCPTT private call | Answer Mode = Manual and and SDP offer **does not contain** any SDP "m=application" media-level section for a media-floor control entity |
| 3 | <-- | MCPTT SIP Core sends SIP 200 (OK) to DUT-A |  |
| 4 | -- > | DUT-B sends SIP 180 Ringing to MCPTT SIP Core |  |
| 5 | -- > | DUT-B sends SIP 200 (OK) to MCPTT SIP Core | Only after the recipient has manually accepted the incoming call |
| 6 | -- > | MCPTT SIP Core sends SIP ACK to DUT-B |  |
| 7 | <-- | MCPTT Server sends MCPC CONNECT to DUT-A |  |
| 8 | <-- | MCPTT Server sends MCPC CONNECT to DUT-B | Inviting User ID of DUT-A is included |
| 9 | -- > | DUT-A sends MCPC ACK to MCPTT Server |  |
| 10 | --> | DUT-B sends MCPC ACK to MCPTT Sever |  |
| 11 | <-- > | DUT-A and DUT-B exchange audio vice versa |  |

## 110.5 Affiliation

### 110.5.1 Affiliation

**Description**

MC service group affiliation can be achieved through the following two methods:

1. Explicit affiliation: An MC service client indicates interest in one or many MC service groups to the MC service server. This interest may be initiated either by an MC service user using the MC service UE, or by an automatic procedure within the MC service client that indicates that the MC service user is interested in the MC service group at that MC service client. An authorized MC service user may remotely modify another MC service user's affiliation to an MC service group.
2. Implicit affiliation: An MC service user's affiliations to MC service groups are determined through configurations and policies within the MC service and performed by the associated MC service server.

This test case describes about explicit affiliation but a DUT to affiliate to a pre-arranged group.

**Applicability**

3GPP Rel. 15 or later

**Related core specifications**

3GPP TS 23.280

3GPP TS 23.379

3GPP TS 24.379

**Reason for test**

To verify the DUT can successfully affiliate to a pre-arranged MC service group.

**Initial configuration**

MCPTT client on DUT-A has been authenticated and authorised but not affiliated (implicitly or explicitly).

MCPTT client on DUT-A has already been provisioned (statically or dynamically) with the group information, or a pointer to the group information, that the MC service client is allowed to be affiliated.

A dispatcher MCPTT client is available and authenticated and authorised to change the affiliation status of MCPTT client on DUT-A.

User has logged in to MCPTT client of DUT-A.

**Test procedure**

**Scenario A: User initiated group affiliation**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | DUT-A sends affiliation request to MCPTT server | DUT-A explicitly requests for a group affiliation.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A and PCAP traces for affiliation parameters. |
| 2 | MCPTT server sends a group affiliation response to DUT-A having checked DUT-A’s authorisation to join the requested group. | DUT-A receives a group affiliation response which contains the group identity and the group’s current state  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A and PCAP traces for affiliation parameters. |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| User initiated group affiliation | | | |
| 1 | --> | DUT-A sends SIP PUBLISH to the server | The MCPTT client shall include MCPTT ID and MCPTT group(s) of interest and the event header field contains the “presence” |
| 2 | <-- | MCPTT server sends SIP 200 OK to DUT-A |  |

**Scenario B: Dispatcher initiated group affiliation**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | The dispatcher client sends affiliation request for DUT-A to MCPTT server | **VERIFICATION METHOD**:   * Visual verification of GUI of the dispatcher client and PCAP traces for affiliation parameters. |
| 2 | MCPTT server sends DUT-A a message to check its availability | **VERIFICATION METHOD**:   * PCAP traces of DUT-A MCPTT client |
| 3 | DUT-A responds with its availability |  |
| 4 | MCPTT server sends a group affiliation response to the dispatcher. | The disptacher receives a group affiliation response which contains the group identity and the group’s current state  **VERIFICATION METHOD**:   * Visual verification of GUI of the dispatcher and PCAP traces for affiliation parameters. |
| 5 | MCPTT server sends group affiliation request to DUT-A | **VERIFICATION METHOD**:   * PCAP traces of DUT-A MCPTT client |
| 6 | DUT-A sends MCPTT group affiliation reponse to MCPTT server. |  |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Dispatcher initiated group affiliation | | | |
| 1 | --> | The dispatcher client sends SIP PUBLISH for DUT-A to the server | The MCPTT client shall include MCPTT ID and MCPTT group(s) of interest abd the event header field contains the “presence” |
| 2 | <-- | MCPTT server sends SIP MESSAGE to DUT-A to check its availability |  |
| 3 | --> | DUT-A responds with SIP 200 OK to MCPTT server |  |
| 4 | <-- | MCPTT server sends SIP 200 OK to the dispatcher client |  |
| 5 | <-- | MCPTT server notifies DUT-A with SIP NOTIFY of the group affiliation request |  |
| 6 | --> | DUT-A sends SIP 200 OK response |  |

## 110.6 Client location submission

### 110.6.1 Client location submission

**Description**

If a location reporting trigger activates, the MCPTT client will check whether the minimum report interval timer is running. If the timer is running the MCPTT client will wait until the timer expires. When the minimum report interval timer expires, the MCPTT client shall, if any of the reporting triggers are still true, send a location information report.

**Applicability**

3GPP Rel. 15 or later

**Related core specifications**

3GPP TS 24.379

**Reason for test**

To verify the DUT can successfully sends a location report.

**Initial configuration**

MCPTT server has configured the MCPTT client on DUT-A to send location report.

MCPTT client on DUT-A has been authenticated and authorised.

**Test procedure**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | DUT-A sends location report when the location report minimum reporting interval timer expires | **VERIFICATION METHOD**:   * PCAP traces have SIP MESSAGE with location information collected on DUT-A and/or MCPTT server logs |
| 2 | DUT-A receives the acknowledgement for the delivery of location report | **VERIFICATION METHOD**:   * PCAP traces for SIP 200 OK collected on DUT-A and/or MCPTT server logs |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Location Information Submission | | | |
| 1 | --> | DUT-A sends SIP MESSAGE to the server | Message contains the current location as the following example:  Content- Type: application/ vnd.3gpp.mcptt- location- info+ xml  ...  < location- info>  < Configuration>  ...  < CurrentLocation>  <CurrentCoordinate> < latitude> 4032351< / latitude> < longitude> 16639587< / longitude> < / CurrentCoordinate>  <MbmsSaId> 2< / MbmsSaId>  <MbsfnArea> 1< /MbsfnArea>  < / CurrentLocat ion>  ...  < / Configuration>  < / location- info> |
| 2 | <-- | MCPTT server sends SIP 200 OK to DUT-A |  |

## 110.7 Ambience Listening Call

### 110.7.1 Remotely Initiated Ambience Listening

**Description**

The ambience listening call is a type of a private MCPTT call that only allows a ”listened to” user to transmit media to a “listening” user such that there is no indication on the MCPTT UE of the “listened to” user about the call and the media transmission.

Remotely initiated ambience listening is started by an authorised user (e.g., dispatcher) who wants to listen to another MCPTT user. In this case, the “listened to” user is the called party, and shall automatically accept the call without causing any indication about the call and transmit the media to the “listening” user.

**Applicability**

3GPP Rel. 15 or later

**Related core specifications**

3GPP TS 23.379

3GPP TS 24.379

**Initial configuration**

DUT-A and DUT-B are authorised to invoke a remotely initiated ambience listening call.

DUT-A is chosen to be the "listening" user, and DUT-B is chosen to be the "listened to" user.

**Scenario A- Remotely initiated ambience listening call using on-demand session**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | DUT-A have the permission to originate a remote initiated ambience listening call | The <allow-request-remote-initiated-ambient-listening> element of the <ruleset> element is present in the MCPTT user profile document and is set to a value of "true" in MCPTT client of DUT-A  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A and/or DUT-A’s MCPTT client configuration parameters. |
| 2 | DUT-A sends remotely initiated ambience listening call request to MCPTT server indicating the “listened-to” user as DUT-B | **VERIFICATION METHOD**:   * Visual verification of GUI and PCAP traces for affiliation parameters of DUT-A |
| 3 | MCPTT server sends the amibience listening call request to DUT-B | DUT-B GUI does not provide any indication of the ambience listening call request arrival.  **VERIFICATION METHOD**:   * Visual verification of GUI and/or PCAP traces of DUT-B MCPTT client |
| 4 | DUT-B ambience listening call response to the MCPTT server |  |
| 5 | MCPTT server sends an ambience listening call response to DUT-A | It indicates that the call is setup successfully.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A and PCAP traces for affiliation parameters. |
| 6 | MCPTT server sends a floor granted to DUT-B and a floor taken to DUT-A | **VERIFICATION METHOD**:   * PCAP traces of DUT-A MCPTT client |
| 7 | The media is transmitted from DUT-B to DUT-A. | **VERIFICATION METHOD**:   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-B should be audible on DUT-A and no indication of an ongoing call on DUT-B |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Remote initiated ambience listening call using on-demand session | | | |
| 1 | --> | DUT-A sends SIP INVITE to MCPTT server | an application/vnd.3gpp.mcptt-info+xml MIME body with the <session-type> element set to a value of "ambient-listening" and  an <ambient-listening-type> element set to a value of "remote-init”;  resource lists contains the MCPTT ID of the “listened-to” user (DUT-B);  and the relevant key material for end-toend security are present in the message. |
| 2 | <-- | MCPTT server sends SIP 183 SESSION PROGRESS to DUT-A | This message is optional |
| 3 | <-- | MCPTT server sends SIP INVITE to DUT-B | an application/vnd.3gpp.mcptt-info+xml MIME body with the <session-type> element set to a value of "ambient-listening" and  an <ambient-listening-type> element set to a value of "remote-init”;  There is no indication of a call arrival on GUI of DUT-B |
| 4 | --> | DUT-B responds with SIP 200 OK to MCPTT server |  |
| 5 | <-- | MCPTT server sends SIP 200 OK to DUT-A | The user is indicated that the call has been successfully established. |
| 6 | <-- | MCPTT server sends DUT-B MCPC Floor granted |  |
| 7 | <-- | MCPTT server sends DUT-A MCPC Floor taken |  |
| 8 | --> | DUT-B sends audio to DUT-A |  |

**Scenario B- Remotely initiated ambience listening call using pre-established session**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | DUT-A have the permission to originate a remote initiated ambience listening call | The <allow-request-remote-initiated-ambient-listening> element of the <ruleset> element is present in the MCPTT user profile document and is set to a value of "true" in MCPTT client of DUT-A  VERIFICATION METHOD: Visual verification of GUI of DUT-A and/or DUT-A’s MCPTT client configuration parameters. |
| 2 | DUT-A sends remotely initiated ambience listening call request to MCPTT server indicating the “listened-to” user as DUT-B | VERIFICATION METHOD:Visual verification of GUI of DUT-A and PCAP traces for affiliation parameters. |
| 3 | MCPTT server sends the amibience listening call request to DUT-B | DUT-B GUI does not provide any indication of the ambience listening call request.  **VERIFICATION METHOD**:   * Visual verification of GUI and/or PCAP traces of DUT-B MCPTT client |
| 4 | DUT-B ambience listening call response to the MCPTT server |  |
| 5 | MCPTT server sends an ambience listening call response to DUT-A | It indicates that the call is setup successfully.  **VERIFICATION METHOD**:   * Visual verification of GUI of the dispatcher and PCAP traces for affiliation parameters. |
| 6 | MCPTT server sends a floor granted to DUT-B and a floor taken to DUT-A | **VERIFICATION METHOD**:   * PCAP traces of DUT-A MCPTT client |
| 7 | The media is transmitted from DUT-B to DUT-A. | **VERIFICATION METHOD**:   * Audio and visual verification on DUT-A and DUT-B * Audio sent by DUT-B should be audible on DUT-A and no indication of an ongoing call on DUT-B |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Remote initiated ambience listening call using pre-established session | | | |
| 1 | --> | DUT-A sends SIP REFER to MCPTT server | an application/vnd.3gpp.mcptt-info+xml MIME body with the <session-type> element set to a value of "ambient-listening" and  an <ambient-listening-type> element set to a value of "remote-init”;  resource lists contains the MCPTT ID of the “listened-to” user (DUT-B);  and the relevant key material for end-toend security are present in the message. |
| 2 | <-- | MCPTT server sends SIP INVITE to DUT-B | an application/vnd.3gpp.mcptt-info+xml MIME body with the <session-type> element set to a value of "ambient-listening" and  an <ambient-listening-type> element set to a value of "remote-init”;  There is no indication of a call arrival on GUI of DUT-B |
| 3 | --> | DUT-B responds with SIP 200 OK to MCPTT server |  |
| 4 | <-- | MCPTT server sends SIP 200 OK to DUT-A | The user is indicated that the call has been successfully established. |
| 5 | <-- | MCPTT server sends DUT-B MCPC Floor granted |  |
| 6 | <-- | MCPTT server sends DUT-A MCPC Floor taken |  |
| 7 | --> | DUT-B sends audio to DUT-A |  |

# 111 Mission Critical Data (MCData)

## 111.1 SDS (Short Data Service), standalone, over SIP

### 111.1.1 One-to-one SDS, standalone, over SIP

**Description**

A MCData user initiates a standalone SDS data transfer with another MCData user. For the SDS data transfer, SIP signalling protocol is used. The target MCData user may be addressed using the functional alias that can be shared with other MCData users.

**Applicability**

3GPP Rel. 15 or later

**Related core specifications**

3GPP TS 23.282

3GPP TS 24.282

**Reason for test**

To verify the DUTs can successfully transfer data over SIP signalling protocol.

**Initial configuration**

The SDS payload data size is below the configured maximum payload data size for SDS over SIP signalling.

MCData clients on DUT-A and DUT-B have been registered for receiving MCData service.

MCData clients on DUT-A and DUT-B belong to the same MCData system.

Optionally, the MCData clients on DUT-A and DUT-B may have activated functional alias to be used.

DUT-A and DUT-B have established end-to-end secuirty context and is active.

**Test procedure**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | DUT-A initiates an SDS data transfer for DUT-B for its consumption | The user of DUT-A compiles the data for DUT-B’s consumption.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A |
| 2 | DUT-A sends a MCData standalone data request towards the MCData server | The data request contains conversation identifier and the DUT-A user includes disposition request for the delivery and read notifications.  DUT-A may include its functional alias and DUT-B’s functional alias within the SDS data transfer  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A and PCAP traces for MCData parameters. |
| 3 | MCData server responds back to DUT-A with a functional alias resolution response message that contains the resolved MCData ID | This step is applicable if DUT-A includes functional aliases in the standalone data request |
| 4 | If the MCData server replies with a MCData functiona alias resolution response message, DUT-A sends a new MCData standalone data request towards the resolved MCData ID. | The data request contains conversation identifier and the DUT-A user includes disposition request for the delivery and read notifications.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A and PCAP traces for MCData parameters. |
| 5 | MCData server sends MCData standalone data request to DUT-B | DUT-B receives MCData payload  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-B |
| 6 | DUT-B sends a MCData data disposition notification of the delivery to MCData server |  |
| 7 | MCData server sends the dispostion notification of the delivery to DUT-A | DUT-A received the disposition notification for delivery.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A |
| 8 | DUT-B sends a MCData disposition notification of the read report to MCData server once the user of DUT-B reads the data. |  |
| 9 | MCData server sends the MCData disposition notification of the read report to DUT-A | DUT-A received the disposition notification for read report  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| one-to-one SDS, standalone, over SIP | | | |
| 1 | --> | DUT-A sends SIP MESSAGE request to the MCData server | DUT-A builds payload message for DUT-B’s consumption and includes MCData ID or the functional alias of DUT-B.  An application/vnd.3gpp.mcdata-info+xml MIME body with the <mcdatainfo> element containing the <mcdata-Params> element with a <request-type> element set to a value of "one-to-one-sds"  The SDS disposition request type is set to “DELIVERY AND READ” |
| 2 | <-- | MCData server sends SIP NOTIFY to DUT-A | This is optional to resolve the functional alias of DUT-B |
| 3 | --> | DUT-A sends SIP MESSAGE request to the MCData server | This is optional only if MCData server sends SIP NOTIFY with resolved functional alias of DUT-B |
| 4 | <-- | DUT-B receives SIP MESSAGE from MCData Server | DUT-B receives a notification of a message delivery, may display the functional alias of DUT-A and render the contents of the payload. |
| 5 | --> | DUT-B sends SIP 200 OK to MCData Server |  |
| 6 | <-- | MCPTT server sends SIP 200 OK to DUT-A |  |
| 7 | --> | DUT-B sends SIP MESSAGE to MCData server | This includes the SDS disposition status response after the user of DUT-B receives and reads |
| 8 | <-- | DUT-A receives SIP MESSAGE that contains the disposition notification | DUT-A is displayed the disposition notification as requested. |

### 111.1.2 Group SDS, standalone, over SIP

**Description**

The initiation of a group standalone SDS to a selected group results in affiliated group members receiving the SDS data. The SDS payload data size is assumed to be below the configured maximum payload data size for SDS over signalling control plane.

**Applicability**

3GPP Rel. 15 or later

**Related core specifications**

3GPP TS 23.282

3GPP TS 24.282

**Reason for test**

To verify the DUT can successfully transfer short data to a pre-arranged group over SIP signalling protocol.

**Initial configuration**

MCData clients on DUT-A, DUT-B and DUT-C have been assigned to the same group and registered for receiving MCData service and affiliated.

MCData cleints on DUT-A, DUT-B and DUT-C belong to the same MCData system.

Optionally, the MCData clients on DUT-A, DUT-B and DUT-C may have activated functional alias to be used.

DUT-A, DUT-B and DUT-C have established end-to-end secuirty context and is active.

The SDS payload data size is below the configured maximum payload data size for SDS over SIP signalling.

**Test procedure**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | DUT-A initiates an SDS data transfer to a pre-configured group along with members DUT-B and DUT-C for their consumption | DUT-A user composes SDS message.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A |
| 2 | DUT-A sends a MCData group standalone data request towards the MCData server | The MCData group data request contains MCData Group ID and the conversation identifier.  DUT-A user includes disposition request for the delivery and read notification.  DUT-A may also include its functional alias within the SDS data transfer.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A and PCAP traces for MCData parameters. |
| 3 | MCData server sends MCData standalone data request to the group to which DUT-B and DUT-C are members | DUT-B and DUT-C receive MCData payload and renders the payload on GUI of DUT-B and DUT-C  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-B |
| 4 | DUT-B and DUT-C send a MCData data disposition notification of the delivery to MCData server |  |
| 5 | MCData server sends the dispostion notification of the delivery to DUT-A | DUT-A receives the disposition notification of delivery from DUT-B and DUT-C as aggregated or individually.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A |
| 6 | DUT-B and DUT-C send a MCData disposition notification of the read report to MCData server once the user of DUT-B reads the data. |  |
| 7 | MCData server sends the MCData disposition notification of the read report to DUT-A | DUT-A received the disposition notification of read report from DUT-B and DUT-C as aggregated or individually  VERIFICATION METHOD:Visual verification of GUI of DUT-A |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| group SDS, standalone, over SIP | | | |
| 1 | --> | DUT-A sends SIP MESSAGE request for the pre-arrnaged group to the MCData server | DUT-A builds payload message for the consumption of DUT-B and DUT-C, and includes MCData Group ID of DUT-B and DUT-C.  An application/vnd.3gpp.mcdata-info+xml MIME body with the <mcdatainfo> element containing the <mcdata-Params> element with a <request-type> element set to a value of "group-sds"  The SDS disposition request type is set to “DELIVERY AND READ” |
| 2 | <-- | DUT-B and DUT-C receive SIP MESSAGE from MCData Server | DUT-B and DUT-C receives a notification of a message delivery, may display the functional alias of DUT-A and render the contents of the payload. |
|  | --> | DUT-B and DUT-C send SIP 200 OK to MCData Server |  |
|  | <-- | MCPTT server sends SIP 200 OK to DUT-A |  |
|  | --> | DUT-B and DUT-C individually send SIP MESSAGE to MCData server. | The message includes the SDS disposition notification to DUT-A after the users of DUT-B and DUT-C receive and read the message |
|  | <-- | DUT-A receives SIP MESSAGE that contains the disposition notification | DUT-A is displayed the disposition notification as requested. |

## 111.2 Enhanced Status

### 111.2.1 Enhanced Status

**Description**

Enhanced status corresponds to information specific to the activities performed by the mission critical service users during their operation(s) (e.g., available, in operation on site, going to the operation site, or just arrived).

The enhanced status information, i.e. the set of possible values corresponding to the activities of the MCData user, shall be configured by the MCData administrator.

**Applicability**

3GPP Rel. 15 or later

**Related core specifications**

3GPP TS 23.282

3GPP TS 24.282

**Reason for test**

To verify the DUTs can successfully share the enhanced status information to the members of a selected group.

**Initial configuration**

MCData clients on DUT-A, DUT-B and DUT-C have been assigned to the same group and registered for receiving MCData service and affiliated.

MCData clients on DUT-A, DUT-B and DUT-C belong to the same MCData system.

The MCData group of DUT-A, DUT-B and DUT-C is pre-configured with the possible values for enhanced status information.

**Test procedure**

|  |  |  |
| --- | --- | --- |
| **-** | **Test procedure** | **Expected behaviour** |
| 1 | DUT-A selects the MCData group and the value of the status information from the list of available pre-configured values to share with the group comprising DUT-B and DUT-C. | DUT-A user selects the group and enahnced status value.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A |
| 2 | DUT-A sends a MCData group standalone data request towards the MCData server | The MCData group data request contains MCData Group ID and the conversation identifier.  DUT-A user includes disposition request for the delivery and read notification.  DUT-A may also include its functional alias within the SDS data transfer.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-A and PCAP traces for MCData parameters. |
| 3 | MCData server sends MCData standalone data request to DUT-B and DUT-C |  |
| 4 | DUT-B and DUT-C receive enhanced status information of DUT-A | DUT-B and DUT-C receive MCData payload and render the operational value as enhanced status to the GUIs.  **VERIFICATION METHOD**:   * Visual verification of GUI of DUT-B |

**Example message flow:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Direction DUT - NW** | **Message** | **Comments** |
| Enhanced status | | | |
| 1 | --> | DUT-A sends SIP MESSAGE request to the MCData server | DUT-A selects operation value from <mcdata-enhanced-status-operational-values> element under <list-service> element and generate a group standalone message and includes MCData Group ID of DUT-B and DUT-C.  An application/vnd.3gpp.mcdata-info+xml MIME body with the <mcdatainfo> element containing the <mcdata-Params> element with a <request-type> element set to a value of "group-sds"  The SDS disposition request type is set to “DELIVERY AND READ” |
| 2 | <-- | DUT-B and DUT-C receive SIP MESSAGE from MCData Server | DUT-B and DUT-C receives a notification of a message and render the operational value as enhanced status to DUT-B and DUT-C |
|  | --> | DUT-B and DUT-C send SIP 200 OK to MCData Server |  |
|  | <-- | MCPTT server sends SIP 200 OK to DUT-A |  |

# 112 Mission Critical Video (MCVideo)

*Editor’s note: not available yet*

Document Management

Document History

| Version | Date | Brief Description of Change | Approval Authority | Editor / Company |
| --- | --- | --- | --- | --- |
| 36 | 20/09/2021 | Rev. of this PRD as described in Annex J, approved at TSG FT#75 | TSG#45 | Momar Goumballe / Orange |
| 37 | 03/01/2022 | Rev. of this PRD as described in Annex J, approved at TSG FT#76 | TSG#46 | Momar Goumballe / Orange |
| 38 | 08/03/2022 | Rev. of this PRD as described in Annex J, approved at TSG FT#77 | TSG#47 | Momar Goumballe / Orange |
| 39 | 18/07/2022 | Rev. of this PRD as described in Annex J, approved at TSG FT#78 | TSG#48 | Momar Goumballe / Orange |
| 40 | 18/10/2022 | Rev. of this PRD as described in Annex J, approved at TSG FT#79 | TSG#49 | Momar Goumballe / Orange |
| 41 | 27/01/2023 | Rev. of this PRD as described in Annex J, approved at TSG FT#80 | TSG#50 | Momar Goumballe / Orange |
| 42 | 8/03/2023 | Rev. of this PRD as described in Annex J, approved at TSG FT#81 | TSG#51 | Momar Goumballe / Orange |
| 43 | 8/07/2023 | Rev. of this PRD as described in Annex J, approved at TSG FT#82 | TSG#52 | Momar Goumballe / Orange |
| 44 | 04/10/2023 | Rev. of this PRD as described in Annex J, approved at TSG FT#83 | TSG#53 | Momar Goumballe / Orange |
| 45 | 08/01/2024 | Rev. of this PRD as described in Annex J, approved at TSG FT#84 | TSG#54 | Momar Goumballe / Orange |

Other Information

|  |  |
| --- | --- |
| Type | Description |
| Document Owner | GSMA Terminal Steering Group |
| Editor / Company | Momar Goumballe, Orange  [momar.goumballe@orange.com](mailto:momar.goumballe@orange.com) |