



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.137

(02/2000)

SERIES T: TERMINALS FOR TELEMATIC SERVICES

**Virtual meeting room management for
multimedia conferencing audiovisual control**

ITU-T Recommendation T.137

(Formerly CCITT Recommendation)

**VIRTUAL MEETING ROOM MANAGEMENT FOR MULTIMEDIA
CONFERENCING AUDIOVISUAL CONTROL**

Summary

This Recommendation defines the functionality required to deliver a combined voice- and data-conferencing (audiographics) service; it is architected to deliver such services using generic and extensible mechanisms that are independent of the underlying transports used for delivering the real-time media streams.

In future revisions this scope is expected to broaden to cover all media services used within a conference.

This protocol extends the Generic Conference Control Recommendation (T.124), adding the capability to define and manage virtual meetings spaces within the GCC conference. It provides a means to seamlessly integrate all of the media components being used within a meeting.

These virtual meeting spaces can be regarded as representing meeting rooms where audio, video and data communication can take place. The functionality provides support for a number of virtual meeting scenarios including round-table meetings, informal meeting spaces and conferences with audiences. The protocol provides inherent support for multi-room scenarios.

The protocol will find application in support of enhanced audio conferencing and audiographics conferencing services. The aim is to provide an holistic approach to the management of the meeting room involving multiple media, while providing the flexibility to adapt individual meeting rooms in response to user requirements. The object of standardizing this functionality is to provide a basis for interoperability between client applications and conferencing services.

Source

ITU-T Recommendation T.137 was prepared by ITU-T Study Group 16 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on 17 February 2000.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 2001

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

	Page
1	Scope..... 1
2	References..... 2
3	Definitions 3
4	Abbreviations..... 4
5	Conventions 5
6	Overview..... 5
7	Meeting room management – Components 6
7.1	Virtual meeting rooms 6
7.2	The lobby 7
7.3	The conference framework 7
7.4	MRM rosters 8
7.4.1	The MRM conference roster..... 8
7.4.2	The MRM room roster..... 8
7.4.3	Roster operation..... 8
7.5	Meeting room types 8
7.6	Meeting room modes 9
7.7	Room Hierarchies and relationships 9
7.7.1	Room access regimes..... 9
7.7.2	Room entry regimes..... 9
7.7.3	Room state 10
7.8	Classes of participation in a conference..... 10
7.8.1	MRM observers 11
7.8.2	MRM participants..... 11
7.8.3	MRM administrators..... 12
7.8.4	MRM groups..... 12
7.9	MRM conference services 13
7.10	Starting an MRM conference..... 13
7.10.1	Private side communication..... 14
7.10.2	Relationship between meeting rooms and data sessions 14
7.10.3	Role and relationship to the GCC conductor..... 15
7.10.4	MRM management nodes..... 15
7.10.5	Proxying MRM functionality..... 15
8	T.137 infrastructure management functionality..... 15
8.1	Use Of MCS channels..... 15
8.2	Static channels 15

	Page
8.3	Dynamic channels 16
9	MRM infrastructure management service definition 16
9.1	Summary of MRM abstract services 16
9.2	Establishing the MRM Top provider 19
9.3	MRM conference framework 19
9.3.1	Description of abstract services 24
9.4	The MRM rosters 26
9.4.1	Description of abstract services 27
9.5	Meeting rooms 32
9.5.1	Description of abstract services 33
9.6	MRM groups 39
9.6.1	Specifying groups in the framework 39
9.6.2	Specifying groups within an active room 39
9.6.3	Description of abstract services 39
9.7	T.120 sessions 43
9.7.1	Description of abstract services 43
9.8	MRM room roles 44
9.8.1	Room role permissions – Specification 46
9.8.2	Room Roles – Description of abstract services 48
9.8.3	The chair role 57
9.8.4	Accessing pre-defined roles 58
9.8.5	Asking a role holder for permission to Act 58
9.9	Permission to take action (Incorporating floor control) 58
9.9.1	Description of abstract services 58
9.10	Conference media controls and indications 64
9.10.1	Description of abstract services 64
9.11	MRM media services 71
9.11.1	Service events 71
9.11.2	Audio Bridging Service 72
9.12	Audio Mixer management 72
9.12.1	Description of abstract services 72
10	MRM protocol definition 79
10.1	MRM node types 79
10.2	Launching MRM 79
10.3	Determining the MRM top provider 79
10.4	MRM initialization and start-up 79
10.5	General Operation 82

10.6	Framework PDUs.....	83
10.7	Roster PDUs	85
	10.7.1 Roster Reporting.....	85
	10.7.2 Updating participants roster information.....	87
10.8	Room PDUs	88
	10.8.1 Creating an MRM Room	88
	10.8.2 Entering An MRM Room	90
	10.8.3 Invitation to an MRM Room	92
	10.8.4 Creating a group	92
10.9	T.120 Session Association.....	95
10.10	Room role PDUs.....	95
	10.10.1 Assigning an MRM role	95
	10.10.2 Relinquishing an MRM role.....	97
	10.10.3 Asking for an MRM role	98
	10.10.4 Passing An MRM Role to another Room Participant.....	98
	10.10.5 Changing role permissions	100
10.11	Permission PDUs	100
	10.11.1 Asking the chair for permission to take action	100
	10.11.2 Relinquishing Permission	101
	10.11.3 Chair grants permission.....	102
10.12	Source Identification PDUs	102
10.13	Indication PDUs.....	103
	10.13.1 Notification of privacy.....	103
10.14	Service Command PDUs	104
10.15	Audio Management PDUs	105
	10.15.1 Service initialization	105
	10.15.2 Audio Mixer Setup	105
	10.15.3 Audio Mixer status reporting.....	106
	10.15.4 Inviting Audio-Only participation	107
	10.15.5 Ejecting an-Audio-Only participant.....	108
11	ASN.1 Definitions	109
12	Requirements for running MRM	129
12.1	Summary of GCC services required by MRM.....	129
12.2	Summary of use of the Multipoint Communication Service	129
	12.2.1 MCS services.....	129
	12.2.2 MCS channels used by MRM.....	130
	12.2.3 MCS Tokens used by MRM.....	131
	12.2.4 Encoding of PDUs in MCS primitives	131

Recommendation T.137

VIRTUAL MEETING ROOM MANAGEMENT FOR MULTIMEDIA CONFERENCING AUDIOVISUAL CONTROL

1 Scope

This Recommendation provides extensions to the Generic Conference Control functionality to support the specification and management of virtual meeting spaces (meeting rooms) within a conference. It provides a cohesive and uniform approach to the management of all media used within the meeting room irrespective of the underlying transport and media encodings used.¹ This work is expected to find application in the provision of commercial conference services.

This Recommendation defines procedures for arbitrating use of real-time media services such as telephony and video streaming², for advertising those services to participating nodes, and for managing and controlling those services once activated. This Recommendation specifies mechanisms for multimedia capability reporting at the start of and during communications, and defines procedures for arbitrating access to, and configuring the infrastructure in support of, the services.

This Recommendation supports conferences in which data communication is established first as well as conferences in which audio communications are established first. The provisions of this Recommendation are independent of the underlying network(s) and the content of real-time streams that traverse those networks. Furthermore, this Recommendation provides the possibility for different types of multimedia system to co-exist within the same conference and for those systems to be controlled in a uniform manner.

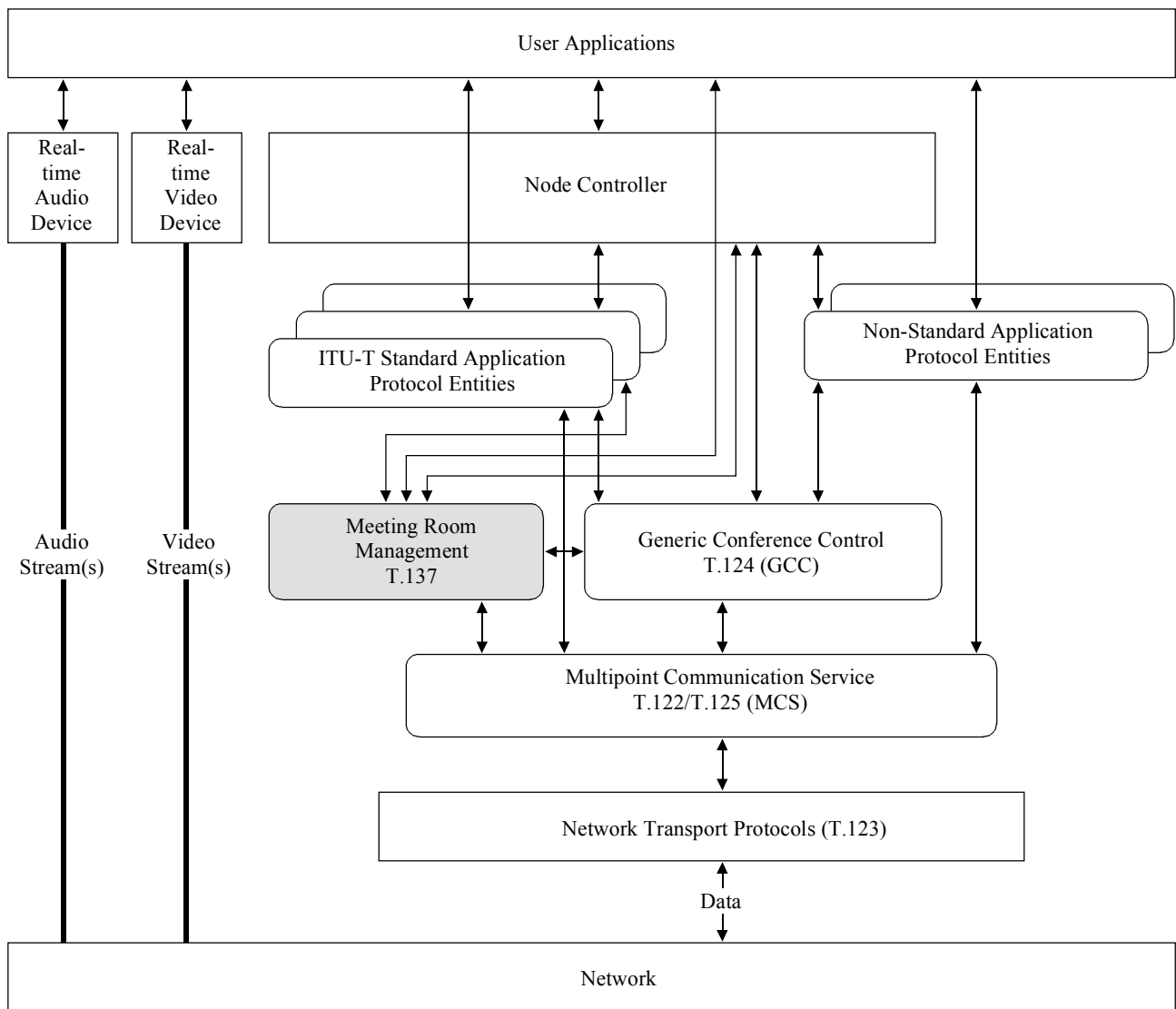
This Recommendation defines a Conference Server that takes responsibility for the running of the MRM Conference and a service access channel for communication between the clients and server. The MRM protocol places minimal implementation burden on client terminals with most of the complexity being concentrated at the Server. This provides centralized management of the real-time infrastructure. A single server node is assigned to be the Conference Manager and the MRM Top Provider. However, communication from clients to the MRM Top Provider is always directed to the Conference Management channel, allowing other Network Elements to also join this channel and to cooperate in delivering the conference. These other Network Elements may shadow the MRM Top Provider, allowing the possibility of one of them taking on the Top Provider role in the event of a failure condition which disables the current Conference.³ This provision also allows the possibility of a distributed management mode in future versions of this protocol, enabling it to more readily scale to support large conference environments.

This Recommendation requires the services of MCS and GCC.

¹ The use of this Recommendation with H-series Recommendations is outside the scope of this work and is part of a separate harmonization activity.

² Video-streaming excludes solutions based upon Recommendation H.323, which are for further study and are outside the current scope.

³ This is currently outside the scope of this Recommendation.



T1607350-99

Figure 1/T.137 – Audiovisual control architecture

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- ITU-T Recommendation T.120 (1996), *Data protocols for multimedia conferencing*.
- ITU-T Recommendation T.121 (1996), *Generic application template*.
- ITU-T Recommendation T.122 (1998), *Multipoint communication service – Service definition*.
- ITU-T Recommendation T.123 (1996), *Network-specific data protocol stacks for multimedia conferencing*.
- ITU-T Recommendation T.124 (1998), *Generic Conference Control*.

- ITU-T Recommendation T.125 (1998), *Multipoint communication service protocol specification*.
- ITU-T Recommendation X.680 (1997) | ISO/IEC 8824-1:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.
- ITU-T Recommendation X.691 (1997) | ISO/IEC 8825-2:1998, *Information technology – ASN.1 encoding rules – Specification of Packed Encoding Rules (PER)*.

3 Definitions

This Recommendation defines the following terms:

- 3.1 application protocol:** Any standard or non-standard protocol specification that makes use of T.120 services.
- 3.2 application protocol entity:** The instantiation of an application protocol in a terminal or MCU.
- 3.3 application protocol session:** A group of peer application protocol entities communicating with each other.
- 3.4 conference framework:** A predefined specification for a conference. The framework is created out of band but is specified in this Recommendation because it is passed into an active conference and its parameters are available through the conference roster.
- 3.5 meeting room model:** A model of a room to support either a formal or an informal meeting between peers. All role holders are supported.
- 3.6 conference room model:** A model of a presentation room which, in addition to the meeting room model, supports Observers.
- 3.7 coffee room model:** An informal meeting space which supports ad hoc communication and dynamic creation and destruction of meeting spaces, these meeting spaces are also represented by MRM rooms.
- 3.8 convenor:** The entity responsible for creation of the MRM conference.
- 3.9 MRM conference:** A conference in which the MRM protocol is used to manage and control all the media used within the conference.
- 3.10 MRM top provider:** Entity within a node responsible for managing an MRM conference.
- 3.11 MRM conference manager:** The name given to the node controller associated with the MRM Top Provider. The conference manager has overall responsibility for running the conference.
- 3.12 MRM room:** A virtual meeting room created within an MRM conference. Effectively, a subset of MRM conference participants formed into a separate and autonomous group of nodes for the purposes of communicating both data and real-time streams within that group.
- 3.13 MRM group:** An association of room participants, allowing them to effectively be treated as a single entity for management, control or information distribution; unlike an MRM room, an MRM group is not an autonomous partitioning of a conference or a room.
- 3.14 MRM terminal:** A terminal node that supports and makes use of the MRM protocol.
- 3.15 MRM participant:** A node which has joined the conference as a GCC conventional node running the MRM protocol and participating in the MRM conference.
- 3.16 MRM observer:** A node which may be either a GCC conventional or anonymous node, running the MRM protocol, but taking an observer role in the conference.

3.17 Multipoint Control Unit (MCU): A device used to interconnect multiple terminals and/or further MCUs to allow multiple participants to be connected to single conference. Also known as "bridge".

3.18 multipoint service: One or more functions that are provided and managed by Network Elements or MCUs.

3.19 network element: The name given to an MRM-capable node that provides management of multipoint conferencing or bridging/processing of real-time streams. MCUs, MCs and gateways are examples of network elements.

3.20 node: Any entity that supports the T.120/T.137 protocols. This includes both terminal equipment and network elements such as routers and MCUs.

3.21 source: The origin of a real-time media stream.

3.22 stream: A flow of real-time information of a specific media type (e.g. audio) and format (e.g. G.722) from a single source to one or more destinations.

3.23 video processor: A device that modifies the content of video streams, e.g. a transcoder, continuous presence processor. (This is primarily a place holder for future use.)

3.24 virtual MCU: Term used to describe the collection of network elements within a real-time conference which present a single logical interface to terminals.

4 Abbreviations

This Recommendation uses the following abbreviations:

APE	Application Protocol Entity
ATM	Asynchronous Transfer Mode
GCC	Generic Conference Control
ISDN	Integrated Services Digital Network
LAN	Local Area Network
MC	Multipoint Controller (as defined in Recommendation H.323)
MCS	Multipoint Communication Service
MCU	Multipoint Control Unit
MP	Multipoint Processor (as defined in Recommendation H.323)
MRM	Meeting Room Management
NSCE	Network Specific Control Entity
PDU	Protocol Data Unit
PSTN	Public Switched Telephone Network
RDC	Remote Device Control
SAP	Service Access Point
VMR	Virtual Meeting Room

5 Conventions

The primitive parameters of the abstract services defined in this Recommendation use the following key:

M Parameter is mandatory

C Parameter is conditional

O Parameter is optional

Blank Parameter is absent

(=) Value of the parameter is identical to the value of the corresponding parameter of the preceding primitive, where preceding is defined relative to the order: request, indication, response, confirm.

(=RQ) Value of the parameter is identical to the value of the corresponding parameter in a preceding primitive, where RQ = request, IN = indication, RS = response, and CF = confirm.

Primitives are categorized in up to four types: Request, Indication, Response, and Confirm. Some primitives support all of these types, while others do not. These four types are defined as follows:

Request primitive: Those that are sourced from a node controller, application protocol entity or user application to initiate a certain action.

Indication primitive: Those that are sourced from an MRM provider either as a result of a Request primitive, or as a result of an MRM-initiated action.

Response primitive: Those that are sourced from a node controller in response to an Indication primitive which is defined to require a response.

Confirm primitive: Those that are sourced from an MRM provider as a result of a Response primitive.

6 Overview

This Meeting Room Management (MRM) Recommendation defines an application protocol that is designed to be used in conjunction with the Generic Conference Control (T.124). MRM provides enhanced conference management and control capabilities that will find application in support of a variety of conferencing scenarios, including audiographics services. Its main application will be in supporting the provision of these conferencing services, where it is designed to provide better simulation of a diverse range of real-world face-to-face meeting scenarios. The Recommendation can be regarded as a toolkit of functionality that can be used to build and configure conferences to match user requirements.

MRM provides the mechanisms to support a number of meeting room types, each with its own set of capabilities. Each room type has an appropriate set of rules specified, relating to participation and facilities available. MRM also standardizes a number of participant roles, assigning privileges and constraints as appropriate for the intended role.

Rooms and roles may be combined flexibly with management regimes to model the meeting requirement and controls and indications are available for all the media to be used within each meeting space. Unlike GCC, the functionality provided is focused around rooms, and there may be more than one virtual meeting room within a GCC conference.

The meeting environment is fully configurable through what is termed the "conference framework", allowing users of the protocol freedom to define their own meeting models using the tools provided. Templates show how the framework may be used to model common meeting scenarios, thus allowing service providers or application writers to easily configure potentially complex requirements by selecting from predefined template models.

This Meeting Room Management Recommendation extends the data management features provided by GCC to include control and management of the real-time aspects (audio and video) of multimedia communication applications. It provides a toolkit of high-level services to control and manage the use of real-time streams within a heterogeneous multipoint communication environment.

The services range from basic tools such as speaker identification and "on-air" indication functions through to services such as Audio mixing and Floor control which can be used to enhance existing audioconferencing services. The protocol provides the basis for delivering more sophisticated multimedia conferencing services in the future. These features are designed to be used in a wide range of group communication scenarios, ranging from the traditional conference (as typically used for business meetings), through to loosely coupled conferences (e.g. social gatherings in which participants may join and leave discussion groups at will). It also supports more complex requirements (e.g. a convention with a panel of participants who interact, watched by a large audience). Provision is made for autonomous communication sessions (termed "rooms") within a conference (e.g. to allow private side conversations or breakout groups), without requiring a separate conference to be established.

The goal of this MRM Recommendation is to provide support for a wide range of real-world conferencing scenarios, thus enabling teleconferencing to be more effectively used as a substitute for face-to-face meetings and conferences, while shielding any resulting complexity, from the end user. By defining a number of conference models to support commonly used meeting scenarios, MRM services can be configured to run automatically with minimal user interaction. Conference participants thus gain the benefits of a range of sophisticated services without the burden of having to understand how to configure or control them.

MRM builds on the strengths of Recommendation T.120 and is thus inherently independent of network and platform. It also overcomes some of the scalability limitations of Recommendation T.120 by allowing use of Anonymous nodes which have minimal interaction with and influence on the conference. Typically, these nodes would be members of an audience.

MRM services apply within GCC-established conference environment. MRM also provides support for conference participants without T.120 to be supported, as in the case of "Audio only" participants.

7 Meeting room management – Components

This MRM Recommendation is realized as T.120 application protocol; it is classified as a Management Application because network element(s) such as MCUs play a key role in the provision of the MRM infrastructure and services specified in the Recommendation. Participants in an MRM conference are however largely unaware of management components involved in provisioning any given conference. Participants communicate through a common management channel to the conference manager, or through a node nominated by the Conference Manager to take on specific responsibilities.

7.1 Virtual meeting rooms

The MRM application protocol has the task of integrating the real-time media components with the data conference and providing a seamless conferencing solution to the user. All MRM protocols operate within the context of what this Recommendation terms a "virtual meeting room"; there may be one or more of these virtual meeting rooms within a GCC conference.

For an MRM conference with only a single virtual meeting room, there is a one-to-one mapping with the GCC conference, and all participants in the GCC conference are participants within the meeting room. However, MRM may also be proxying for non-T.120 nodes.

Additional virtual meeting rooms effectively provide subgroupings of the GCC conference participants. For each virtual meeting room, the MRM protocol provides separate management tools for all the media components to be used within that virtual meeting room.

MRM defines a single default meeting room and allows additional rooms to be defined as required.

Additional meeting rooms may be predefined using the conference framework mechanisms. These rooms are known as "persistent" rooms as they exist for the duration of the conference irrespective of whether there are any participants actually in the room or not. Meeting rooms may also be dynamically created during the conference. The creator may choose if these rooms are to persist or not. By default these rooms are public and are listed in the MRM Room roster. Private virtual meeting rooms may also be created; these do not appear in the roster: they are dynamically created when required and the creator invites the desired participants.

MRM supports the association of application protocols with a particular meeting room. Once an application protocol is associated with a room, it is only available to the participants of that room.

7.2 The lobby

The lobby is a defined space inside the MRM conference but outside of any MRM meeting rooms. Its purpose is to provide a place to discover what rooms are available or a space to create rooms. For participants who already know their room destination, the lobby is transited through on the way to their destination room. New entrants to the lobby receive an MRM conference roster that contains a room list.

7.3 The conference framework

The MRM conference framework consists of a comprehensive set of parameters that allow definition of all aspects of an MRM conference, allowing such aspects as scale, management and access regimes and even the look of the conference to be determined.

The framework allows the creator of the conference (the convenor) the capability to very precisely model a real-world conferencing requirement as a teleconference event.

The MRM protocol has a default framework configuration built in to it; this is used as a template for the conference, unless a new framework is loaded. It is not necessary to specify a whole new framework in order to adjust individual parameters. Only when framework parameters are defined are they used to override the default framework settings. A "convenor"-defined framework may thus contain as few or as many of parameters as may be required to effect the desired configuration.

The MRM conference framework consists of three parts. The first part contains the conference configuration template that holds conference wide parameters. The second part contains room constructor templates, and the third part contains service constructor templates.

The conference configuration may also specify a list of predefined rooms, each specified room being given a unique handle within the framework. The parent room is also specified to allow its relationship with other specified rooms to be understood. Each room may also be given a name and a purpose description.

Room constructors allow the convenor to specify each room type and the permitted roles within each. Room access controls, governing the participation of a node in a particular role, may also be specified by the convenor. Each defined role may have a role access key specified, allowing the intended role holder, on entry to the conference, to be identified and assigned the intended role. These MRM access keys are defined within room constructors and communicated out of band to the intended role holder(s). When an access key holder joins an MRM conference and provides an MRM access key, the MRM top provider is responsible for allocating that node the correct role or level of participation in the conference room.

Nodes can be associated into groups, for the purpose of control, with each group being assigned a common access key. An ordered group may be defined to provide, for example, a sequence of presenters. Each member of the group is given a position in the sequence and unique key.

By using the conference framework, detailed models and regimes can be put in place, allowing the conferencing protocols to be configured for a diverse range of applications without burdening the users of the protocol with significant additional complexity. An MRM conference framework may consist of only a single parameter to be overridden, or it may contain a completely new model and management regime for the MRM conference.

7.4 MRM rosters

The MRM roster is a database of configuration and status information that is held and maintained by the MRM top provider for the duration of an MRM conference. Other network elements may shadow the Top Provider role where robust implementations are required. The MRM roster mechanism is independent from the GCC roster. It is designed explicitly to support the needs of the MRM conference protocol. The roster itself consists of two parts: the conference roster and the room roster. The node controller associated with the Top MRM provider acts as the MRM conference manager and is responsible for instructing the Top MRM Provider what information to include in its roster reports.

7.4.1 The MRM conference roster

The MRM conference roster contains conference-wide information, relating to the MRM conference and its configuration. This will include framework parameters, either the defaults, or any parameters resulting from a framework-load that have been used to configure the conference. The conference roster can also contain a list of available or active rooms. Each entry corresponds to a room roster where details of individual room configurations and status are stored.

7.4.2 The MRM room roster

The MRM room roster contains information on each of the rooms that have been instantiated in a particular conference. For each room listed there will be information about its configuration and operation mode. The associated MRM database also acts as a depository for any room constructor templates that have been defined. For each room listed there will be a state and a mode parameter and also an access regime. Room state will be either inactive if there are no participants, or active if there is one or more participants. The room mode will be either permanent, or persistent, or dynamic.

7.4.3 Roster operation

When a node first enters an MRM room, it gets sent a personal copy of the conference roster and also a personal copy of the room roster for the room it is entering. A node entering a room must immediately join the room communication channel specified in order to get room roster updates, which provide incremental changes to the original roster it received. Nodes may also make individual requests of the roster. The roster itself is configurable, allowing the content of roster reports and roster updates to be determined from the specification in the framework, or as a result of a node asking for information through the roster-enquire mechanism.

7.5 Meeting room types

This Recommendation defines one standard meeting space, termed the "default meeting room". Provision is also made to allow users to define their own meeting room types. These meeting room types are designed to identify specific types of operational model. They may be considered as handles to identify predefined room constructor templates.

MRM also defines the lobby, a virtual space inside the MRM session, but outside any MRM meeting room. A Conference communication channel is associated with the lobby and used to distribute

conference-wide information. The Lobby is not an MRM room and therefore it does not support other media or data sessions. Any node in an MRM session but not in an MRM room is located in the lobby.

7.6 Meeting room modes

Three meeting room modes are currently supported:

Permanent rooms – Rooms of this type are hardwired into the protocol. Currently the protocol only supports one permanent room, which is used as the default room for the conference. In the case of there only being a single meeting room for the conference, then this room is used and is directly equivalent to the GCC conference.

Dynamic Rooms – Dynamic rooms are created and destroyed as required during the conference.

Persistent Rooms – A persistent room is a special type of dynamically created room that the top provider creates in response to a user requirement or a framework specification. The Top provider joins this room in a purely administrative role to maintain its persistence from its creation time and for the duration of the MRM conference. The MRM top provider does not appear in the room roster for that room.

7.7 Room Hierarchies and relationships

When a room is created, it may be created as a base level room, or it may be created as a child of an existing room. Child rooms would for example be used for private side communication or for breakout sessions. These child rooms should be regarded as separate virtual spaces within the parent room. If a room is dynamic, then the child must also be dynamic. If a room is persistent, the child may be persistent or dynamic. If the parent is permanent, then the child room may be permanent, persistent or dynamic. When a parent is destroyed, so is the child. A dynamic room is not regarded as empty if it has child rooms which are occupied. The default configuration of the MRM protocol supports a depth of two in the room hierarchy; this allows side communication to take place. The framework parameter for room hierarchy must be set to one to prohibit side communication or to a value greater than two to allow additional levels of hierarchy for more complex scenarios.

7.7.1 Room access regimes

Currently, the protocol supports three standardized access regimes.⁴ The creator of a room should specify which is required. In the absence of this parameter the protocol will assume the room is to have "open" access. The supported regimes are:

Open – The default, no admission controls, room appears in MRM-Roster's room list and participants are free to join and leave the room at will.

Controlled – Access is via a password or key. The room appears in the MRM-Roster's room list.

Private – Access is by invitation only. The room details required to gain entry to a room do not appear in the roster, but an indication of all those nodes in private session is obtainable from the roster. Participants who enter by invitation will be returned to the room or place they came from when they leave, unless that room no longer exists, in which case they are placed into the default room.

7.7.2 Room entry regimes

Private rooms have an entry regime built in that only allows entry by invitation. Open and controlled rooms may be entered directly. However, MRM allows the entry regime of a room to be modified. A room may be configured with an entry regime that does not allow direct entry, but requires entry via

⁴ MRM admission controls are room based and are independent of the GCC mechanisms.

an automated reception procedure, or via a staffed reception. To avoid the creator being locked out by a prescriptive entry regime, a master key is always created for a room when it is created and it is given to the room creator. This key always provides direct access to that room and can be used to override any other entry requirements. The operator and convenor of the conference also have access to master keys.

Nodes attempting to enter the room with an indirect entry regime operating and without supplying the master key will be placed in a private reception room. This private space may be used by some form of automated exchange (which is beyond the scope of this Recommendation) or there may be direct interaction with a "door keeper". The role of "door keeper" is by default held by the operator, but may be assigned to a room or conference participant. A room may have one or more "door keepers" assigned to receiving new entrants. A node wishing to enter the room would attempt to enter the room directly. This would result in a "Permission-Ask" being generated and directed to a "door keeper" for the room. The "door keeper" in responding to the request and the incoming node are placed in private conference in the reception room allowing registration, validation, or other procedures required for entry to take place. The "door keeper" has to issue an invitation to that person to get them into the main room.

7.7.3 Room state

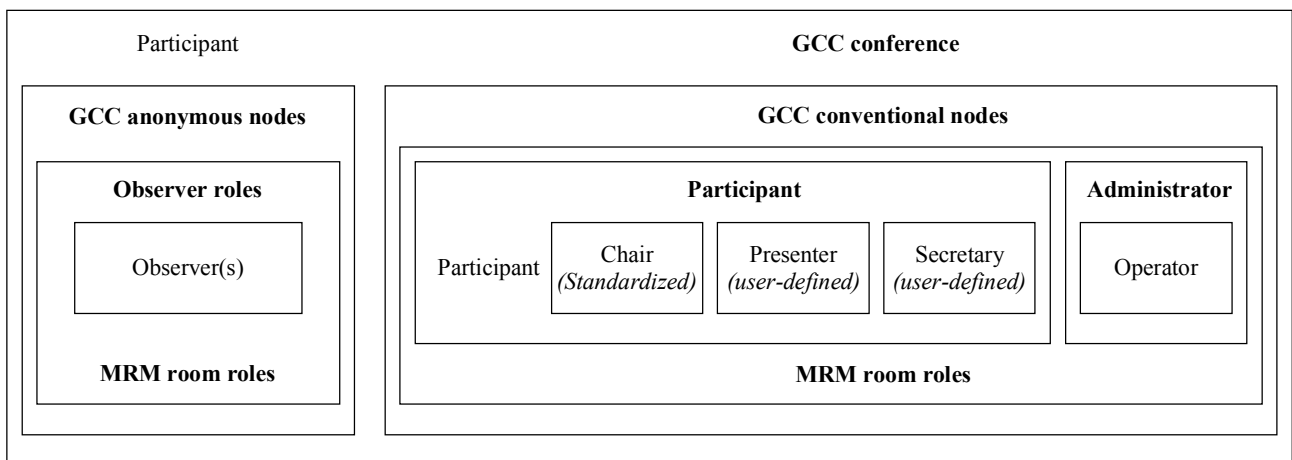
Each MRM room has a state parameter associated with it. Once a room has been created, it takes on the state "created"; once a room is ready to accept entering participants, it enters the "ready" state. When the room is occupied, it enters the "active" state. If a room is terminated before the end of a conference, it maintains an entry in the roster with the state "destroyed". The "suspended" state is a placeholder for future use.

7.8 Classes of participation in a conference

GCC supports two levels of participation in a T.120 conference: "Conventional nodes" and "Anonymous nodes". MRM utilizes both of these and builds upon them within the MRM protocol.

Figure 2 identifies three role categories, which are supported by the MRM protocol:

- Observer roles;
- Participant roles;
- Administrator roles.



T1607360-99

Figure 2/T.137 – Relationship between MRM roles and GCC node classifications

The observer role maps directly onto a GCC anonymous node, while the participant and administrator roles map onto the GCC conventional node classification. A participant role holder may also take on additional role responsibilities by taking on Chair, Presenter, or Secretary responsibilities as appropriate.

7.8.1 MRM observers

The role of MRM observer has a direct equivalence to a GCC anonymous node. The "observer" class is provided to allow participants who are primarily observers to join conference and enter rooms. This class can be considered as a special role. Support or exclusion of observers may be specified in advance in the conference framework. When a terminal joins a conference it may specify it wishes to join as an observer, or its invitation may contain conference and room access parameters which only allow access and participation as an observer. The observer role is automatically assigned to a node enrolling as a GCC Anonymous node in the conference. It is anonymous to the other participants in that it does not appear in the conference or room rosters. However, the conference can be configured so the chair/convenor/secretary gets notification of an observer joining and leaving. An observer is primarily a viewer and this role can be used as an aid to scalability or as a means for restricting input to the conference. Typically, observers would not be able to participate interactively in a conference unless explicitly granted permission through use of the floor control service. Observers are by default able to see and hear the exchanges within the room that they are in. The MRM protocol supports the capability for managed input by an observer to the proceedings within a room. Where permitted, an observer could make a request to the chair to ask a question or to use a data application. This classification provides support for the observer requirement as well as supporting participation scalability for conferencing applications. One example of appropriate use of the observer role would be in support of a delegate at a convention.

7.8.2 MRM participants

An MRM participant is a conventional T.120 node and is able to fully participate and interact within the GCC conference. MRM inherits the GCC conventional node capabilities and allows MRM participants to participate and interact fully within a meeting room. A participant has access to, and may take on, additional supplementary roles which bring added privileges and responsibilities; for example, the chair role makes the role holder the focal point of the floor control protocol. The MRM conference manager provides a proxy for non-T.120 nodes such as "audio-only" participants, allowing them to participate as "audio-only" participants.

7.8.2.1 Supplementary conference room roles

This MRM Recommendation defines a number of regular participant roles which may be used to accord particular conference participants with additional privileges and status with respect to use of MRM services. Users may also define their own roles using the privilege tables provided by MRM. Every role has an associated set of rules to support its intended use. MRM services are aware of the active roles within a room and will modify their behaviour to accommodate the requirements and privileges. The protocol allows roles to be preassigned and held by nodes in an inactive state until needed; alternatively they may be dynamically acquired. Unlike the GCC conductor role, MRM roles are used within the context of a meeting room and have a scope covering all the media used within that meeting space.

7.8.2.1.1 Chair role

The chair role is provided to allow a node to become the chairperson for a particular meeting room. The chairperson is a participant and has responsibility for managing the floor-control procedures within that room. Where multiple rooms are permitted, each room may have its own chair role. The MRM chair for the base room may be aligned with the GCC conductor by synchronizing MRM with the data session.

The chair has the following privileges available to it:

- Activation/deactivation of floor control;
- Operation of floor control;
- The ability to invite/eject audio-only participants – *plus the ability to determine if they join direct or via a private conference with chair or an entry lobby;*
- The ability to invite/cancel invitation to an operator – *Either audio only or audio with data.*

7.8.2.1.2 Convenor role

The convenor of an MRM controlled conference is also the GCC convenor. It would typically be the convenor who defined any conference framework associated with the conference. Conferences are convened using GCC primitives that make explicit provision for MRM parameters.

7.8.3 MRM administrators

The MRM protocol currently supports only one administrator role: that of the "operator". The operator role is provided to support conference offered as part of a managed service. Where permitted, the role can be taken on by an external management/helper in order to support/educate and problem-solve, for conference participants.

The operator joining a conference takes on this special system role of operator. It is a predefined role that specifies how a service operator can interact with a conference meeting room session. The privileges associated with this role are:

- An operator cannot be muted and will be mixed by the audio bridge (*where audio is present*).
- The operator can take on the chair role if needed and can eject participants.
- The operator may have access to data content for the room restricted.

There is no protocol limit to the number of operators who may join a conference or enter a virtual room. This is a local matter for the service provider.

7.8.4 MRM groups

The MRM protocol supports the ability to associate participants to form MRM groups. The purpose of a group may be specified explicitly or determined by the framework that is in use. A group can be used to manage communication and information distribution between group members who are participating in a conference. Groups are primarily designed to be used within the context of a room, but where appropriate and with caution they may be allowed scope across rooms within a single conference. MRM groups, like rooms, form a partitioning of conference members; however, this is not for separate and autonomous communication. The group is simply a means to associate participants within the room. The group mechanism aims to provide support for a diverse range of requirements. Each group that is created has its own private communication channel.

Two standard group types are supported. The first type of group is termed a "team". A room may support many teams. The chair protocol is aware of groups and can interact at a group level, for example giving a group permission to speak rather than an individual. A team may be used to represent a panel of experts or one team in a competition. Teams may be used to group companies together in a meeting or distinguish between locations. The other type of group supported is an "ordered" group; this may be used to represent an ordering of presenters for an event, or responsibility for topics on a meeting agenda. Each member of an ordered group is assigned a sequence number. Where an ordered group of presenters is defined, the role when released, automatically moves down the sequence to the next in the group.

Groups can be used as the context for services and roles within a selected conference framework. Group management functionality provided by MRM includes group create/disband, group join/leave

(allowing unregulated group access) and group add/remove node (allowing an authorized node to control group membership). A group membership list is held by the MRM top provider in the room roster and is available to terminals on request. Within a group the following functions would be supported in respect of the roles: the ability to assume, relinquish and assign a role, plus the ability to activate and deactivate a role holder.

7.9 MRM conference services

MRM conference services are blocks of related functionality provided by one or more network elements to the MRM conference. MRM supports the standardization of services, an example of this being the audio bridging service. MRM also supports the user-defined services which can be negotiated in the MRM application capabilities as the application enrolls.

An MRM conference has a standard default conference framework hard-wired into the protocol to ensure that basic interoperability can be achieved; this default serves as a fallback operational mode and specifies baseline functionality that must be supported in order to be MRM compliant. By default, no MRM conference services are required. The MRM conference service provider is able to modify virtually all of the default framework parameters, and, by loading a new framework specification, the functionality available can be further restricted or enhanced. When a framework is loaded into a conference, the MRM defaults are overridden and replaced by the new settings. A service provider is able to set a framework flag Update Default Framework, which causes the framework loaded outside a specific conference to be stored and used in preference to the standard default.

Each MRM service will have an operational requirement and an operational state. Each MRM service will have one of the following operational requirements assigned to it: "Required", "Offered", "Restricted", "Prohibited". The standard default framework does not list any MRM services. If an application capability exchange identifies capability, this is appended to the room roster for the nodes. The capability will itself provide the operational requirement for a given service. If a new framework is loaded, then this may contain a list of MRM services each of which will have an operational requirement. Any service list provided by a framework will become part of the conference and room rosters as appropriate. When a node joins a room it receives a room roster which tells it which services are Active, and of those that are active, which are required and which are offered but are optional, and also which services (if any) are inactive but are available for activation.

Authority to issue service commands to activate and subscribe to services is also determined by the framework. Even where a service is required, this requirement only applies for nodes capable of supporting the media being used by the service; e.g. an audio node would not be required to support a video-based service.

7.10 Starting an MRM conference

When a GCC conference is created to support an MRM application, the node controller at the GCC top provider is responsible for launching the MRM application protocol. This node also becomes the MRM top Provider and its node controller takes on the role and responsibilities of the MRM Conference Manager.⁵ The MRM Top provider takes on the creation and management of the MRM rosters. The Top MRM provider holds a default configuration for an MRM conference, effectively a default framework, built into the protocol. This default framework allows the creation of additional rooms and roles. Loading a new framework at the time the conference is created allows default settings to be modified. Participants entering the conference will receive the MRM roster directly

⁵ If the Top GCC provider does not support MRM, it is possible for another network element to take on the role, but this is not the recommended mode of operation.

from the MRM conference manager, which contains information on the conference configuration as well as information on the rooms available within the conference.

When a participant enters a room, he will receive a room roster from the top provider; the framework then determines what is permitted within that room.

7.10.1 Private side communication

By default, only a single room will exist and all terminals participating in an MRM-managed conference will interact with each other within a single MRM communication session, which the MRM protocol represents as a Virtual meeting room. However, in some scenarios, there may be a requirement for separate autonomous audiovisual and data communications within the same conference. For example, in a business meeting there may be a requirement for private side conversations, whilst a social gathering may consist of a number of discussion groups which participants join according to their interest in the topic of conversation. MRM supports these requirements by allowing multiple communication spaces (virtual meeting rooms) to be established within a single T.124 conference. MRM rooms are advertised using MRM's own roster mechanisms.

7.10.2 Relationship between meeting rooms and data sessions

An MRM conference consists of a single MRM application session in which one or more virtual meeting rooms are established. MRM services are provided in the context of Virtual Meeting Rooms. The MRM application session is realized as a T.124/T.121 application protocol session. MRM also supports child rooms (which have no direct T.124 equivalent), allowing side conferencing or separate communicating groups to exist within a meeting space. When a node joins an MRM session it will, by default, enter the default meeting room, transiting through the lobby and receiving a conference roster, unless a framework access key causes it to be directed to another room. On entering a room, a participant is automatically subscribed to (and may thus make use of) the active services within that room. Inactive services must be subscribed to explicitly. The interpretation of subscribing a node to a service is service specific.

The MRM protocols are designed to provide support in managing a diverse range of teleconference scenarios. To facilitate this, MRM provides a number of commonly used conference models, each tailored for use in a particular type of meeting environment. When creating a conference, a convenor may select a conference model appropriate for the desired scenario. The requirements of this model are then conveyed to other participants through the definition of a conference framework – a set of parameters – which effectively defines a management regime under which the conference operates. User-defined conference models may also be used if required; the conference convenor simply selects an appropriate set of framework parameters, which can also be used to create new templates allowing for reuse.

The conference framework philosophy allows the conference convenor to establish potentially complex conference scenarios, without exposing participants to the complexity of configuring or using the associated conference services. The provided conference models simplify the task of the convenor for commonly required scenarios, and also greatly simplify the task of repeating a conference based upon templates already defined.

MRM defines three general classifications for group communication rooms: the Meeting Room Model, the Conference Room Model, and the Coffee Room Model. These classifications are used to label a set of parameters that configure an MRM room in support of a particular application requirement. If these basic models are not sufficient, then users of the protocol are free to provide their own parameter groupings in order to tune the room configuration and management regime to meet their own requirements.

7.10.3 Role and relationship to the GCC conductor

Some existing data applications make use of the GCC conductor: for the default room the MRM chair and GCC conductor are aligned. When multiple rooms are used, the MRM room chair mechanisms are required and the GCC conductor is no longer relevant.⁶

7.10.4 MRM management nodes

The MRM top provider is the node responsible for managing the MRM conference. Other management nodes may assist the top provider in delivering the conference. Participants are shielded from conference administration issues as they always communicate to the conference management through a common management channel. Typically, management nodes are network elements such as MCUs.

7.10.5 Proxying MRM functionality

Pre-MRM terminals will still be able to benefit from MRM functionality where another node (network element) is able to proxy on its behalf. Proxying may also be used to support receive-only terminals.

The realization of a proxy agent is a local matter. The problem of knowing where to send control and indications destined for a proxied terminal is handled by T.120. The proxy agent uses the GCC mechanisms to obtain a User ID for its terminal(s). MCS can then correctly route management and control information directed to that terminal, irrespective of the topology.

It is also feasible to allow MRM protocols to arrive from some out-of-band mechanism and be introduced into the conference by a proxy agent that is proxying for T.120. This again is a local matter. An example of this would be a convenor needing to interact with an ongoing conference from a reservation system/terminal. Another example would be a basic audio/video-only terminal that had no data or conference management requirements, but could still benefit from the functionality of an MRM-managed conference.

8 T.137 infrastructure management functionality

8.1 Use Of MCS channels

A single MCS channel – the "Room Participant Communications" channel – represents the virtual meeting room. This MCS channel ID is used to provide a unique identifier for the room. When a room is instantiated from a framework, the pre-assigned room handle will be associated with the MCS channel identifier for that room and both will then be accepted as valid identifiers for that room.

For a multipoint conference with just the default meeting room, the MRM application utilizes three MCS static channels to provide management and control services in support of the MRM conference. Each additional room requires an MCS dynamic channel for participant communication and, optionally, a second dynamic channel for observer communication when required.

8.2 Static channels

MRM-Conference-Management-Channel – It is joined by the top MRM provider and may also be joined by other network nodes that are cooperating to delivering MRM services to the conference. Participating nodes send to this channel in order to communicate with the service management node running the MRM conference. The participants themselves do not need to be aware of which node is actually running the conference; they always send to this well known static channel ID, which

⁶ Changes in GCC and or data apps may be required to fully exploit the capabilities of the MRM protocol.

provides a standard way for communication, independent of the conference topology. It effectively allows client terminals to see the network as a single virtual network element. This mechanism also allows other network elements to shadow or even take over the role of MRM top provider without the need to tear down the conference.

MRM-Conference-Communication-Channel – It is joined by the MRM provider at terminal nodes when they enrol active in the MRM session. It is used to deliver the conference roster and to broadcast information to all MRM participants, irrespective of which room they are in.

MRM-Default Room-Participant-Communication-Channel – All client nodes wishing to participate in the default room join this channel, and it is used to distribute information within the room. The top MRM Provider uses this channel to broadcast to the room. GCC conventional nodes that have entered the room as participants may also send to this channel in accordance with the MRM protocol rules.

MRM-Default-Room-Observer-Communication-Channel – It is only required when a room has observers taking part. It may be used by the top MRM provider to send status and other broadcast information targeted at the observers in the room. A separate channel is used to allow partitioning of different levels of information and thus aids scalability for conferences, which have many observers involved.

8.3 Dynamic channels

In addition to the static channels, the MRM application makes use of one or more dynamic channels in the support of each additional meeting room, which mirror the static communication channels for the static room.

MRM-Room-Participant-Communication-Channel – Each additional meeting room requires its own dynamically allocated MCS channel which is used to broadcast status updates to all the participants in that room. Each room may also have a pre-assigned identifier defined within the framework and outside of the protocol.

MRM-Room-Observer-Communication-Channel – Each additional meeting room may also require its own dynamically allocated MCS channel for observer communications when a room is required to support observers. It may be used by the top MRM provider to send status and other broadcast information targeted at the observers in the room. As with the static provision, a separate channel allowing the partitioning of different levels of information and thus aiding scalability for conferences which have many observers involved.

9 MRM infrastructure management service definition

9.1 Summary of MRM abstract services

Table 1 is a list of all MRM infrastructure management primitives and their associated PDUs. The table also shows whether or not each primitive is mandatory (M), conditionally required (C), or optional (O) for an MRM terminal or for an MRM network element. The table also shows, for the corresponding PDUs, whether these are mandatory (M) or conditionally required (C) for both the transmit (T) and receive (R) directions. A conditionally required PDU is one, which is required only if the corresponding primitive is to be supported at that node. Where a primitive is not mandatory, but its corresponding PDU is mandatory, this implies that there is some portion of the protocol not related to the primitive that relies on the use of that PDU which is required to be supported.

Table 1/T.137 – MRM infrastructure management primitives and PDU

Function Unit	Primitives	Observer Terminal	Participant Terminal	Network Element	Associated PDUs	Direction	Observer Terminal	Participant Terminal	Network Element
Framework	MRM-Launch request	-	-	M	-	-	-	-	-
	MRM-Launch confirm	-	-	M	-	-	-	-	-
	MRM-Framework-Load request	-	-	M	-	-	-	-	-
	MRM-Framework-Load confirm	-	-	-	-	-	-	-	-
	-	-	-	-	FrameworkLoadIndication	T	-	-	M
	-	-	-		FrameworkLoadIndication	R	-	-	M
MRM Rosters	MRM-Roster-Report request	-	-	M	RosterReportIndication	T	-	-	M
	MRM-Roster-Report indication	O	M	M	RosterReportIndication	R	O	M	M
	MRM-Roster-Update indication	-	M	O	RosterUpdateIndication	T	-	-	M
					RosterUpdateIndication	R	-	M	M
	MRM-Roster-Enquire request	O	O	O	RosterEnquireIndication	T	O	O	O
					RosterEnquireIndication	R	-	-	M
Room Management	MRM-Room-Create request	-	O	M	RoomCreateRequest	T	-	O	M
					RoomCreateRequest	R	-	-	M
					RoomCreateResponse	T	-	-	M
					RoomCreateResponse	R	-	C	M
	MRM-Room-Enter request	M	M	O	RoomEnterRequest	T	M	M	M
	MRM-Room-Enter indication	-	-	M	RoomEnterRequest	R	-	-	M
	MRM-Room-Enter response	-	-	M	RoomEnterResponse	T	-	-	M
	MRM-Room-Enter confirm	M	M	O	RoomEnterResponse	R	M	M	M
				RoomReadyIndication	T	M	M	M	
				RoomReadyIndication	R	-	-	M	
	MRM-Room-Invite request	-	O	O	RoomInviteIndication	T	-	O	M
	MRM-Room-Invite indication	-	M	M	RoomInviteIndication	R	M	M	M
MRM Groups	MRM-Group-Create request	-	O	M	GroupCreate Request	T	-	O	M
					GroupCreateRequest	R	-	-	M
	MRM-Group-Create confirm	-	C	M	GroupCreateResponse	T	-	-	M
					GroupCreateResponse	R	-	C	M
	MRM-Group-Disband request	-	O	M	GroupDisbandRequest	T	-	O	M
					GroupDisbandRequest	R	-	-	M
	MRM-Group-Disband confirm	-	C	M	GroupDisbandResponse	T	-	-	M
					GroupDisbandResponse	R	-	C	M
Service Support	MRM-Group-Membership-Specify request	-	O	M	GroupMemberSpecifyRequest	T	-	O	M
					GroupMemberSpecifyRequest	R	-	-	M
					GroupMemberSpecifyResponse	T	-	-	M
					GroupMemberSpecifyResponse	R	-	C	M
	MRM-Service-Command request	-	O	M	ServiceCommandRequest	T	-	O	M
	MRM-Service-Command indication	-	-	M	ServiceCommandRequest	R	-	-	M
	MRM-Service-Command confirm	-	C	M	ServiceCommandResponse	T	-	-	M
					ServiceCommandResponse	R	-	C	M
Session Association	MRM-Session-Associate request	-	O	M	SessionAssociateRequest	T	-	O	M
	MRM-Session-Associate indication	-	-	M	SessionAssociateRequest	R	-	-	M
	MRM-Session-Associate confirm	-	O	M	SessionAssociateResponse	T	-	-	M
					SessionAssociateResponse	R	-	O	M

Table 1/T.137 – MRM infrastructure management primitives and PDU (concluded)

Function Unit	Primitives	Observer Terminal	Participant Terminal	Network Element	Associated PDUs	Direction	Observer Terminal	Participant Terminal	Network Element
Room Role Assignment	MRM-Room-Role-Assign request	-	O	M	RoomRoleAssignRequest	T	-	O	M
					RoomRoleAssignRequest	R	-	-	M
					RoomRoleAssignResponse	T	-	-	M
	MRM-Room-Role-Assign confirm	-	O	M	RoomRoleAssignResponse	R	-	O	M
	MRM-Room-Role-Release request	-	O	M	RoomRoleReleaseRequest	T	-	O	O
					RoomRoleReleaseRequest	R	-	-	M
					RoomRoleReleaseResponse	T	-	-	M
	MRM-Room-Role-Release confirm	-	O	M	RoomRoleReleaseResponse	R	-	O	O
	MRM-Room-Role-Please request	-	O	-	RoomRolePleaseRequest	T	-	O	-
	MRM-Room-Role-Please indication	-	O	M	RoomRolePleaseRequest	R	-	M	M
					RoomRolePleaseResponse	T	-	M	M
	MRM-Room-Role-Please confirm	-	O	-	RoomRolePleaseResponse	R	-	O	-
	MRM-Room-Role-Give request	-	O	-	RoomRoleGiveRequest	T	-	O	O
	MRM-Room-Role-Give indication	-	O	M	RoomRoleGiveRequest	R	-	O	M
				RoomRoleGiveResponse	T	-	O	M	
MRM-Room-Role-Give response	-	O	M	RoomRoleGiveResponse	R	-	O	O	
MRM-Room-Role-Give confirm	-	O	-	RoomRoleGiveIndication	T	-	O	M	
				RoomRoleGiveIndication	R	-	O	M	
MRM-Role-Permissions-Modify request	-	C	M	RolePermissionsModifyRequest	T	-	C	M	
				RolePermissionsModifyRequest	R	-	-	M	
				RolePermissionsModifyResponse	T	-	-	M	
MRM-Role-Permissions-Modify confirm	-	C	M	RolePermissionsModifyResponse	R	-	C	M	
Permission To Act	MRM-Permission-Ask request	O	M	-	PermissionAskRequest	T	O	M	-
	MRM-Permission-Ask indication	-	M	-	PermissionAskRequest	R	-	M	-
	MRM-Permission-Ask confirm	O	M	-					
	MRM-Permission-Relinquish request	O	M	-	PermissionRelinquishRequest	T	O	M	-
	MRM-Permission-Relinquish indication	-	M	-	PermissionRelinquishRequest	R	-	M	-
	MRM-Permission-Relinquish confirm	O	M	-					
	MRM-Permission-Grant request	-	M	M	PermissionGrantRequest	T	-	M	-
MRM-Permission-Grant indication	O	M	M	PermissionGrantRequest	R	M	M	-	
MRM-Permission-Grant confirm	-	M	M						
Source Identification	MRM-Source-Identification request	O	M	M	SourceIdentificationRequest	T	O	M	M
	MRM-Source-Identification indication	O	M	M	SourceIdentificationRequest	R	O	M	M
	MRM-Source-Identification confirm	O	M	M					
	MRM-On-Air-Indication request	O	M	M	OnAirIndicationRequest	T	O	M	M
MRM-On-Air-Indication indication	O	M	M	OnAirIndicationRequest	R	O	M	M	
Privacy	MRM-Privacy-Notify request	-	O	M	PrivacyNotifyRequest	T	O	M	M
	MRM-Privacy-Notify indication	-	O	M	PrivacyNotifyRequest	R	O	M	M
					PrivacyNotifyResponse	T	-	-	M
	MRM-Privacy-Notify confirm	-	O	M	PrivacyNotifyResponse	R	-	-	M
Audio Mixing	MRM-Audio-Mix-Report request	-	-	M	AudioMixReportRequest	T	-	-	M
	MRM-Audio-Mix-Report indication	M	M	M	AudioMixReportRequest	R	-	M	M
	MRM-Audio-Mixer-Set request	-	M	M	AudioMixerSetRequest	T	-	M	M
	MRM-Audio-Mixer-Set indication	-	-	M	AudioMixerSetRequest	R	-	-	M
					AudioMixerSetResponse	T	-	-	M
MRM-Audio-Mixer-Set confirm	-	M	M	AudioMixerSetResponse	R	-	M	M	
Telephony Support	MRM-Invite-Audio-Only-Node request	-	O	M	InviteAudioOnlyNodeRequest	T	-	O	M
	MRM-Invite-Audio-Only-Node indication	-	-	M	InviteAudioOnlyNodeRequest	R	-	-	M
	MRM-Invite-Audio-Only-Node response	-	-	M	InviteAudioOnlyNodeResponse	T	-	-	M
	MRM-Invite-Audio-Node-Only confirm	-	O	M	InviteAudioOnlyNodeResponse	R	-	O	M
	MRM-Eject-Audio-Only-Node request	-	O	M	EjectAudioOnlyRequest	T	-	O	M
	MRM-Eject-Audio-Only-Node indication	-	-	M	EjectAudioOnlyRequest	R	-	-	M
	MRM-Eject-Audio-Only-Node response	-	-	M	EjectAudioOnlyResponse	T	-	-	M
	MRM-Eject-Audio-Only-Node confirm	-	O	M	EjectAudioOnlyResponse	R	-	O	M
	MRM-Call-Status request	-	C	M	CallStatusIndication	T	-	-	M
MRM-Call Status indication	-	-	M	CallStatusIndication	R	-	C	M	

9.2 Establishing the MRM Top provider

Where the MRM protocols are used as part of a managed service, the MRM top provider should be co-located with the GCC and MCS top providers, the location of which may be defined by the service. In other cases where arbitrary allocation of top provider can occur, MRM provides an optional procedure using an MCS Token to determine which node takes on the MRM top provider role.

An MRM provider intending to participate as a Network Element and that is capable of becoming the MRM top provider should check the state of a *local* flag to determine if it needs to compete for the top provider role. This flag may be hardwired, or set as part of the node's configuration, but this is a local matter. Without *a priori* knowledge, nodes should by default be configured to compete for the role. Where this mechanism is used, nodes attempt to grab the MCS token assigned to the MRM top provider (MRM-Token-0). The node that is successful is required to become the MRM top provider, unless a request is made for the token by a node higher in the tree.

In all cases it is highly desirable that the MRM top provider should be co-located with the GCC/MCS top provider; however, this mechanism also allows the possibility of the MRM top provider being at a different node. Although this is not recommended and it may adversely affect the performance of the protocol, neither is it prohibited by the protocol. This is because the possibility exists that the GCC top provider may not support MRM. Where such cases arise, the MRM top provider should be as close as possible to the top of the tree hierarchy.

Creation of an MRM controlled conference

When a GCC conference requiring MRM functionality is created, the conference manager at the top provider launches the MRM application protocol and at the same time tells the provider if it intends to load a new framework. In cases where a new framework is to be loaded, the MRM top provider will be required to wait for it to be loaded before proceeding to enrol the application protocol entity.

Client nodes joining the GCC conference see the availability of MRM functionality through the GCC application roster. Nodes that support the MRM application should, at this stage, enrol their MRM Application Protocol Entity. Enrolment results in that node being sent a personal copy of the MRM conference roster. Nodes enrolled in the MRM conference arrive in the lobby, a defined space outside of any MRM room. They must then use the room entry procedures to enter an MRM room. Where there is more than one room in a conference, nodes may, subject to permission, use the conference roster room list to find and enter rooms. When a node enters a room, it will also receive the roster for that room.

The use of MRM services ideally requires MRM to be present at the top provider node and all network elements within the conference. Terminals are not required to support MRM in order for the MRM protocol to operate, although its absence is likely to give reduced functionality. Network elements may choose to proxy MRM functionality for non-MRM capable terminals; how this is achieved is outside the scope of this Recommendation.

9.3 MRM conference framework

Table 2 below defines the conference framework parameters that may be used in an MRM-Framework-Load primitive. The table also shows the default values for each. Every network element that is capable of becoming an MRM top provider holds a copy of these default values, and in the absence of a new framework, uses them when setting up an MRM conference.

The framework supports many parameters, allowing a great deal of flexibility to configure an MRM conference. However, virtually all of these framework parameters are optional, and typically a new framework will contain only a few parameters needed to modify the default configuration to meet the user's requirements. This is because in the absence of a parameter, the top provider will continue to use the default framework value.

The "New Default Framework" flag allows the possibility for the convenor to tell the MRM top provider to use a new framework to permanently replace the standard defaults. This helps in the situations where the convenor always requires a particular framework configuration from an MRM top provider. The conference framework consists of three parts. The first part contains the conference specification template – a set of parameters that may be used to specify conference-wide parameters. One of these parameters provides a place to list any room constructors needed to realize rooms for the conference. The second part of the conference framework holds the room constructors definitions that the conference specification refers to. Another of the parameters contains a list of predefined rooms; each predefined room must be based upon the default or a room constructor defined within the framework. Constructors may be defined that are not used to create predefined rooms. These constructors serve as room templates and can be made available to participants, for use in a conference. The conference specification also includes a list of services available to the conference. Each service is either based upon the default configuration for that service or a Service constructor that will be defined in the third part of the conference framework.

Table 2/T.137 – MRM conference framework parameters

Conference Framework					
PARAMETER NAME	Description	Range	Unit	Default	Key
Conference Name	Human-readable name for conference	–	String	Null	O
Conference Description	Human-readable description for conference	–	String	Null	O
Conference Model	User-defined parameter		Key	0	O
Max Conventional Nodes Permitted	Enforced limit on number of participants	0-8191		128	O
Max Anonymous Nodes Permitted	Enforced limit on number of observers	0-16 383		512	O
Permitted Services List	List of services permitted	–	Service IDs	Audio	O
Permitted Roles List	List of roles which may be used	–	Role IDs	Chair	O
Synchronize with Data Conference Flag	Flag to indicate if data sessions should get role information	–	Flag	FALSE	O
Groups Permitted Flag	Flag to indicate if groups may be used	–	Flag	FALSE	O
Prohibited functions	List of functions that participant nodes may not send	–	List of PDUs	None	O
Max number of Rooms	Limit of the number of rooms in conference	1-255	Integer	16	O
Max Depth of Rooms		1-7	Integer	2	O
Number of Predefined Rooms	Indicates how many room constructors are to follow	0-255	Integer	1	O
Predefined Room List consisting of:					
Room Handle	Framework's handle for room	0-65 535	Integer	0	O
Room Constructor ID	ID for room constructor to be used	0-65 535	Integer	0	O
Room Parent	Framework handle for parent room	0-65 535	Integer	NULL	O
Room Name	Name of room		String	NULL	O
Room purpose	Human-readable description of room purpose		String		

Table 2/T.137 – MRM conference framework parameters (concluded)

Conference Framework					
PARAMETER NAME	Description	Range	Unit	Default	Key
Participant Conference Roster Reporting parameters	Use Detailed Reports	–	Boolean	FALSE	O
	Include Configuration	–	Boolean	FALSE	O
	Include Status	–	Boolean	TRUE	O
	Include Participants List	–	Boolean	FALSE	O
	Include Participants Details	–	Boolean	FALSE	O
	Allow Roster Enquiry	–	Boolean	FALSE	O
Participant Room Roster Reporting parameters	Use Detailed Reports	–	Boolean	FALSE	O
	Include Configuration		Boolean	FALSE	O
	Include Status		Boolean	TRUE	O
	Include Participants List		Boolean	TRUE	O
	Include Participants Details		Boolean	FALSE	O
	Include room Templates		Boolean	FALSE	O
Observer Conference Roster Reporting parameters	Use Detailed Reports	–	Boolean	FALSE	O
	Include Configuration	–	Boolean	FALSE	O
	Include Status	–	Boolean	TRUE	O
	Include Participants List	–	Boolean	FALSE	O
	Include Participants Details	–	Boolean	FALSE	O
Observer Room Roster Reporting parameters	Use Detailed Reports	–	Boolean	FALSE	O
	Include Configuration	–	Boolean	FALSE	O
	Include Status	–	Boolean	TRUE	O
	Include Participants List	–	Boolean	TRUE	O
	Include Participants Details	–	Boolean	FALSE	O
	Include room Templates	–	Boolean	FALSE	O
Room Constructor List	See Table 3	–	–	Default room only	O
MRM Service Framework	See Table 4	–	–	No services	O

Table 3/T.137 – MRM conference framework – Room constructors

Room Constructor					
Parameter	Description	Range	Unit	Default	Key
Room Constructor Handle	Unique identifier for constructor	0-65 535	Integer	0	M
Framework Room ID	Unique identifier for room	0-65 535	Integer	0	C
Room Name	Human-readable string	–	String	Null	O
Room Purpose Descriptor	Human-readable string	–	String	Null	O
Room Type	Choice of Lobby, Meeting Room, etc.		Enumerated	Default	O
Room Mode	Permanent, Persistent or Dynamic		Enumerated	Persistent	O
Room Parent	Framework Room ID	–	0-65 535	0	O

Table 3/T.137 – MRM conference framework – Room constructors (concluded)

Room Constructor					
Parameter	Description	Range	Unit	Default	Key
Room Management Regime	No chair/chair available/chair active, etc.	–	Enumerated	Chair Available	O
Room Management Style	Informal/Formal/Loosely coupled, etc.	–	Enumerated	Informal	O
Room Media	Choice of one or more of the following: Audio, Video, Data, Real-time data	–	Enumerated	Audio, Data	O
Room Access Model	Choice of Open, Controlled, or Private	–	Enumerated	Open	O
Room Entry Regime	Enter Direct/Via chair/Via lobby	–	Enumerated	Direct	O
Room Exit Regime	Exit to previous/to lobby/exit conf	–	Enumerated	To Previous	O
Child Rooms Permitted	Flag indicating if child rooms are allowed	–	Flag	TRUE	O
Max Participants Permitted	Enforced limit on participants in room	0-16 383	Integer	128	O
Max Observers Permitted	Enforced limit on observers in room	0-16 383	Integer	512	O
Participant Access Key	A password assigned in framework	–	Password	Null	O
Observer Access Key	A password assigned in framework	–	Password	Null	O
Groups Permitted	Flag indicating if group functionality is available	–	Flag	FALSE	O
Chair Required Flag	Flag indicating if chair is required	–	Flag	FALSE	O
Chair Access Key	A password assigned in framework	–	Password	Null	O
Operator Required Flag	Flag indicating if operator is required	–	Roles IDs	FALSE	O
Operator Access Key	A password assigned in framework	–	Password	Null	O
Groups Required List	List of named groups	–	Group IDs	FALSE	O
Group Access Key(s)	List of groups access keys	–	Password	FALSE	O
Audio Support Required flag		–	Flag	TRUE	O
Audio only support		–	Flag	TRUE	
User-defined Role Specification					O
Floor control Required Flag		–	Flag	FALSE	O
MRM Service list: – Service Identifier – Service Requirement – Participation Requirement – Observer Requirement					O

Table 4/T.137 – Service framework – Audio bridging parameters

Audio Bridging – Service Constructor						
Parameter name	Scope	Description	Range	Unit	Default	Key
Service Constructor Handle		A standard key for standardized services and a non-standard key for User-defined services	–	Key	0	M
Audio Mode	Terminal	Mono, Stereo, Spatial	–	Enumerated	Mono	O
Max Number of Participants per node	Terminal	MRM allows more than 1 audio participant per node	1-512	Integer	1	O
Max Number of Audio channels per participant	Terminal	Support for stereo, or multi-channel audio	1-8	Integer	1	O
Max Number of Mixer ports required per conference	Network element	1 channel connects to 1 port	3-512	Integer	3	O
Permissions to Control Mix	Network element	Flag	–	Boolean	FALSE	O
Start-Up Mode	Network element	Choice of Voice-switched or Manual	–	Enumerated	Automatic	O
Mixing Management Mode	–	Centralized Distributed Manual	–	Enumerated	Centrally	O
Number of separate mixer session (rooms)	–	The number of mixers determines how many separate rooms the conference can support	1-127	Integer	1	O
Floor Control	–	Flag		Boolean	TRUE	O O

Audio Mode: This parameter defines the audio mixing mode required.⁷ The parameter supports:

- **Max Number of Audio channels per participant:** This parameter defines the maximum number of audio channels any individual participant may provide to the conference.
- **Max Number of Participants per node:** This Parameter defines the maximum number of participants that can be supported at an individual node.
- **Max Number of Mixer ports required per conference:** Each mixer port can accommodate a single mixer channel. This parameter gives the total number of ports available to the conference.

Start-up Mode: Manual/Automatic.

Mixer Reporting flag: Flag enables or disables Mixer status announcements, triggering roster updates.

Permissions to Control mix: List of nodes/roles permitted to control the service.

Number of separate mixer sessions supported: Number from 0 to *n*.

NOTE – *n* should not exceed half the expected number of participants in the room and is typically less.

Floor Control: A flag indicating if floor control is supported.

⁷ The algorithms and mechanism required to mix in the supported modes are beyond the scope of this Recommendation.

9.3.1 Description of abstract services

- MRM-Launch – Used by the convening Node Controller at a Network Element to initialize its local MRM Provider and to indicate the intention to load a new framework should it become MRM Top Provider.
- MRM-Framework-Load – Used by the Conference Manager to load a new framework (provided by the convenor) at the MRM Top Provider.

9.3.1.1 MRM-Launch

MRM-Launch is used by the node controller at the convening node to indicate to its MRM provider that it should initialize and if the "Compete for Top Provider role" flag is set, it should compete for the MRM top provider token. If it is successful in obtaining the MRM top provider token, it should then check the "New Framework Load Flag". If this is set to TRUE, it should wait for a new framework to be loaded before proceeding to act upon the framework requirements.

Figure 3 shows the sequence of events when using this primitive. Table 5 shows the parameters and types of this primitives.

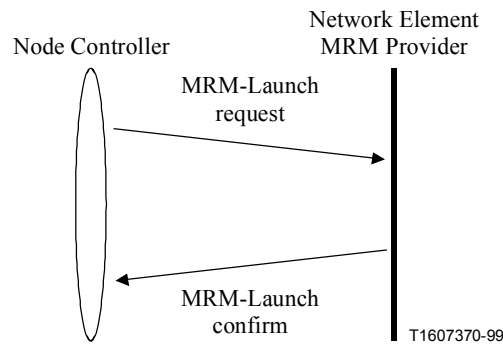


Figure 3/T.137 – MRM-Launch – Sequence of primitives

Table 5/T.137 – MRM-Launch – Types of primitives and their parameters

Content	Request
New Framework Load Flag	M
Compete for Top Provider Role Flag	M

New Framework Load Flag: When flag is set, this tells the MRM provider that if it takes on the role of MRM top provider it must wait for a new framework to be loaded before enrolling the MRM application.

Compete For Top Provider Role Flag: If this flag is set, it tells the MRM provider it should compete for the MRM top provider role by trying to grab the MRM-Top Provider-Token.

9.3.1.2 MRM-Framework-Load

Protocol link: 10.6

The MRM protocol has a default framework configuration built into it, and an MRM-Framework-Load is not required in order to start an MRM conference. A new framework need only be loaded by the convenor when one or more of the default parameters needs to be changed. A new framework only needs to contain the actual parameters that are to be changed. Where a parameter is not supplied, the default will continue to be used.

The MRM conference is created locally at the MRM top provider. If a new framework is to be loaded, it must be done immediately after conference creation and before any other nodes may join. The convenor may load a new framework by passing it to the conference manager (but this is a local matter). The conference manager is then responsible for generating an MRM-Framework-Load request, containing the new framework specification. The MRM top provider acknowledges receipt of a new framework by sending a confirm message to the conference manager. Once this has occurred, the top provider proceeds with the MRM conference establishment.

On receipt of an MRM-Framework-Load request primitive the MRM provider shall check that it holds the MRM top provider token (MRM-Token-0) and that it has previously received an MRM-Launch-Indication asking it to wait for a new framework to be loaded. If these conditions are met, the MRM provider shall take the parameters contained in the new framework and use them to replace the default parameter values. If any of the parameters are out of range or incorrect, the result is locally set to "invalidParameters"; otherwise, the result is set to "successful". If the criteria are not met for receiving a new framework, the result will be set to "notAuthorized". The MRM provider generates an MRM-Load-Framework confirm primitive containing the result and sends it to the Control MRMSAP. The MRM Provider then has to act upon the framework requirement to initialize the MRM conference. The MRM top provider actions are specified in 10.4 (MRM initialization and start-up).

MRM provides a second form of the MRM-Framework-Load function that allows the MRM top provider the ability to relay the framework on to other network elements, thus allowing them to shadow the top provider, or to cooperate in delivering the MRM conference.

Figures 4 and 5 shows the sequence of events when using this primitive. Table 6 shows the parameters and types of this primitives.

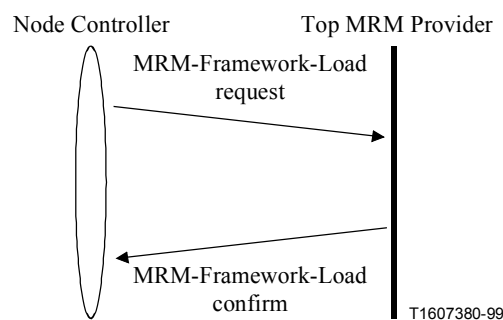


Figure 4/T.137 – MRM-Framework-Load – Sequence of primitives

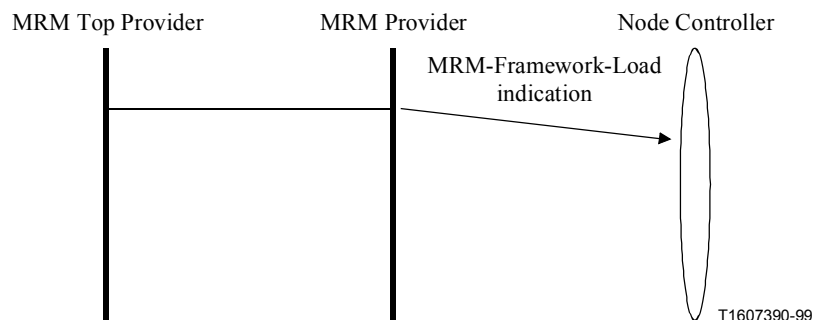


Figure 5/T.137 – MRM-Topology-Load In band distribution to network elements – Sequence of primitives

Table 6/T.137 – MRM-Topology-Load – Types of primitives and their parameters

Content	Request	Indication	Confirm
Request Handle	M	–	M(=)
New Default Framework Flag	M	M	
Originator Handle	O	O	–
Framework Handle	O	O	–
Conference Framework	M	M	
Result			M

Request Handle: Locally unique number to allow the request to be matched with the confirm.

New Default Framework Flag: When this flag is set, the framework contained in the primitive should be used to replace the default framework.

Originator Handle: This is the ID handle assigned to its creator by the service; this is an optional parameter that allows the creator to signal its identity. Predefinition of a conference framework occurs out of band and this parameter identifies that out-of-band entity to the service, unlike the MCS node ID of the convener which only identifies the entry point and may not allow resolution of who the out-of-band convener is.

Framework Handle: This is the ID handle assigned to the framework by its creator; this is an optional parameter.

Conference Framework: For parameters, see the framework specification in Table 4.

Result: A response from the MRM top provider indicating whether the request was accepted or rejected, and, if rejected, the reason why. It contains one of the following of possible results: Successful, Not authorized, Invalid framework parameters.

9.4 The MRM rosters

The MRM roster mechanism is independent from the GCC roster. It is designed explicitly to support the needs of the MRM conference. The MRM roster consists of two parts: the MRM conference roster and the MRM room roster. The MRM conference roster holds information applicable to the whole conference while the room roster holds information about specific rooms. Both MRM rosters are held and maintained by the MRM top provider. The node controller associated with the top MRM provider has the role of MRM conference manager. As conference manager it can use the

MRM-Roster-Enquire request primitive to obtain specified parameter updates from the MRM top provider.

9.4.1 Description of abstract services

Following is a list and a summary for each of the primitives defined in this subclause:

- MRM-Roster-Report – Used by the top provider to deliver a conference roster to a node entering a conference and a room roster to a node entering an MRM room.
- MRM-Roster-Update – Used by the conference manager to deliver an incremental roster update to all participants of a particular room or a conference.
- MRM-Roster-Enquire – Used by an MRM participant or observer to request specific information from the MRM roster.

9.4.1.1 MRM-Roster-Report primitives

Protocol link: 10.7.1

Normally, roster reports are initiated by the MRM top provider as a result of a node joining an MRM conference, entering an MRM room or in response to an enquiry.

The default conference framework specifies what information should be included in a top provider initiated roster report, for a room report or for a conference report. If a new framework is used for a conference, the content of the reports can be modified and also be dependant on the level of participation of the recipient node.

A roster report may also occur as a result of a roster enquiry. the enquiry is able to specify the desired content of the report.

The conference manager may use the MRM-Roster-Report request primitive when it needs to initiate an unsolicited roster report; in such a situation all its parameters are conditionally required. This request primitive may be used to force a roster report when automatic reporting is disabled, or when providing only very limited information. This level of flexibility is primarily provided to support scalability; the recipients of the report are determined by the specified delivery channel or channels. Other nodes may request specific roster information using the MRM-Roster-Enquire mechanism.

Figures 6 and 7 show the sequence of events when using this primitive. Table 7 shows the parameters and types of this primitives.

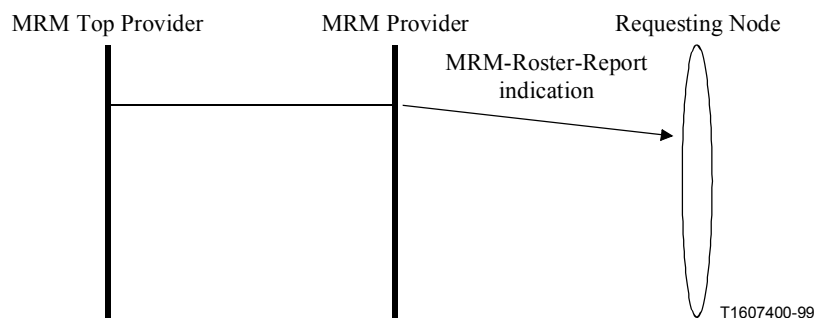


Figure 6/T.137 – MRM-Roster-Report: Automatic Reporting or in response to MRM-Roster-Enquire – Sequence of primitives

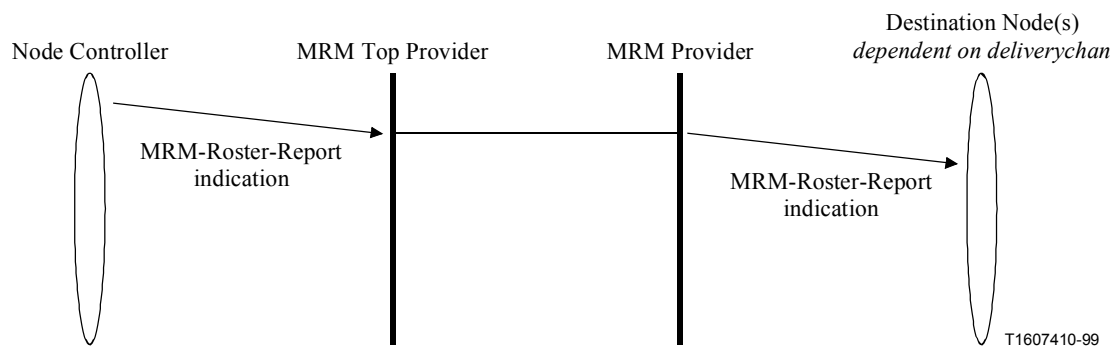


Figure 7/T.137 – MRM-Roster-Report: Conference Manager initiated – Sequence of primitives

Table 7/T.137 – MRM-Roster-Report – Types of primitives and their parameters

Content	Request	Indication
Channel ID List – determining Roster Recipient(s)	M	–
Use Defaults Flag	C	–
Scope of Report	C	–
Include Configuration Information – Flag	C	–
Include Status Information– Flag	C	–
Include Participants List – Flag	C	–
Include Participants Details – Flag	C	–
Include Rooms List – Flag	C	–
Include Information on other Rooms – Flag	C	–
Roster	–	M

Channel ID: Specifies which channel the report should be sent on. For a conference it would be the MRM-Conference-Communications-Channel, for a room it would be the Room ID, for a group it would be a Group channel ID, for an individual it would be the MRM User ID. This parameter may also include a list of User IDs.

Use Defaults Flag: When this flag is set, the other flags are ignored and the conference framework determines which parameters are sent in the report.

Scope of Report: This determines if report is a Conference Report, a Room report, a Group Report, or an Individual Report.

Include Configuration Information: This flag tells the MRM Top provider to include configuration information.

Include Status Information: This flag tells the MRM top provider to include status information.

Include Participants List: This flag tells the MRM top provider to include a participant list.

Include Participants Details: This flag tells the MRM top provider to include participants details.

Include Rooms List: This flag tells the MRM top provider to include a room list.

Include Information on other Rooms: This flag tells the MRM Top Provider to include information on all the other rooms in the conference roster report.

Roster Information: The contents of a roster report PDU is determined by the default roster specification. The default Roster Parameter Set may be modified as a result of a Roster Enquire with

the flag settings in a RosterEnquireIndication PDU, or by an MRM-Roster-Report request telling the MRM top provider what information to include.

The MRM room roster may also include:

Room Roster Information: (All parameters are optional)

Room Handle: A conference-wide unique handle by which the room may be identified and accessed.

Roster Sequence Number: A unique index number issued by MRM every time a new roster Report or Update is sent.

Conference Configuration Information

ConfName: Name of the conference.

ConfDescription: Text string describing purpose of the conference.

ConfModel: *Choice of conference modes – Unspecified, Formal meeting, Informal meeting, Multi-room conference, Broadcast conference, collaboration, gathering or user-defined. These parameters have no impact on protocol; they simply communicate an indication of requirements that may be interpreted locally.*

MaxConventionalNodes: The maximum number of conventional nodes permitted in the conference.

MaxAnonymousNodes: The maximum number of conventional nodes permitted in the conference.

PermittedServicesList: List of MRM services permitted in conference.

PermittedRolesList: List of MRM roles permitted in the conference.

GroupsPermittedFlag: Flag indicated if groups are permitted in the conference.

MaxRooms: The maximum number of room permitted in the conference.

NumRoomsDefined: Number of rooms predefined for the conference.

ProhibitedPDUsList: List of PDUs that will not be permitted for the conference.

Conference Status Information

Number of Participants: The number of participants currently in the conference.

Number of Observers: The number of observers currently in the conference.

Number of Rooms: The number of rooms currently defined within the conference.

Room Depth: The current depth of rooms in the conference.

Room List: List of all the defined rooms by Room ID.

Conference Participants list: List of all the conference participants.

Room Configuration:

Room ID: A unique identifier for a specific room.

Room Framework ID: The Framework ID which identifies a room constructor used to specify the room (when the room is predefined).

Room Mode: The room determines scope and lifetime of the room.

A permanent room is a room defined within the protocol, while a persistent room is dynamically created room, created and made persistent by the MRM top provider. A dynamic room only exists while the creator is joined to the room. A private room is owned by its creator and admission is only through invitation from the creator.

Room Management Model: Parameter identifies a management model to support a given scenario.

Room Management Style: Parameter indicates the level of formality required.

Room Type: The room type is a usage classification; a number of types are defined within the Recommendation: lobby, meeting room, presentation room, gathering, personal room, team conference, panel conference. These types provide and communicate a classification of proposed use, but the precise interpretation is a local matter.

Room Media: This parameter specifies which media the room is required to support. A set selected from the following: Audio, Video, Data, Realtime Data.

Room Access Model: This parameter specifies if the room is to be open, or if access is to be controlled, or if the room is private and entry is to be via invitation only.

Room Entry Regime: This specifies the procedure for entry to the room: either direct, via chair, via lobby or by invitation.

Room Exit Regime: This specifies what happens when a node leaves a room: it may go back either to the previous room it was in or to the lobby, or it may exit the conference.

Room Hierarchy: This parameter specifies where this room is in the hierarchy by identifying its parent.

Required Roles List: A list of the roles required within this room.

Required Services List: A list of the services required in this room and their current status.

Room Status

Room State: The current state of the room – choice of ready/active/suspended.

Number of participants: Actual number of participants in the room.

Number of Observers: Actual number of observers in the room.

Active Roles List: List of roles active within the room.

Active Services List: List of services active within the room.

Associated Application Protocol Sessions: List of application protocol session associated with the room and only available to participants of that room.

Associated Groups List: List of groups active within the room.

Room Participants list: List of participants within the room. *Plus audio-only participants.*

Participants details: Record of participant details (if available): User Handle, Name, Company, Address, email, Telephone No., Fax No.

9.4.1.2 MRM-Roster-Update primitives

An MRM-Roster-Update indication is sent by the top MRM provider. There are two cases where the MRM-Roster-Update mechanism is used. The PDU is issued in response to any event in the room that causes an update of a room roster parameter that was included in the original roster report. The roster update is broadcast to all nodes that have joined the room's communication channel. It contains only incremental changes from the last update or from the room report. The actual parameters sent will be based upon a default list of parameters specified in the protocol. This may only be overridden by using the framework mechanism. All updates and reports contain a sequence number, to ensure old information is never interpreted as being new, due to race condition between roster reports and roster updates. The MRM-Roster-Update indication is also used to provide conference roster updates on the conference communication channel.

Figure 8 shows the sequence of events when using this primitive. Table 8 shows the parameters and types of this primitives.

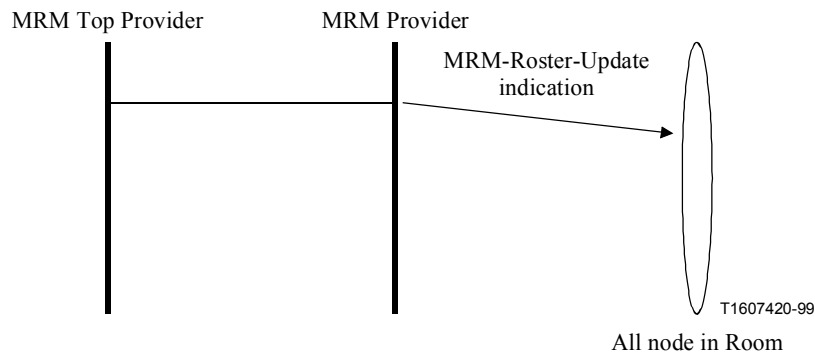


Figure 8/T.137 – MRM-Roster-Update – Sequence of primitives

Table 8/T.137 – MRM-Roster-Update – Types of primitives and their parameters

Content	Indication
Room Configuration Information	O
Room Status Information	O
Room Participants List	O
Conference Configuration Information	O
Conference Status Information	O
Participants List	O
Participant Details	O

Room Configuration Information: (see "Roster Report" above for parameter definitions)

Room Status Information: (see "Roster Report" above for parameter definitions)

Room Participants List: List of participants in the room.

Conference Configuration Information: (see "Roster Report" above for parameter definitions)

Conference Status Information: (see "Roster Report" above for parameter definitions)

Participants List: List of participants in the conference.

Participants Details: Record of participant details.

9.4.1.3 MRM-Roster-Enquire

The MRM-Roster-Enquire is used by a node to request specific information about the room or conference from the MRM top provider. The Enquire will be answered with a roster report.

Figure 9 shows the sequence of events when using this primitive. Table 9 shows the parameters and types of this primitives.

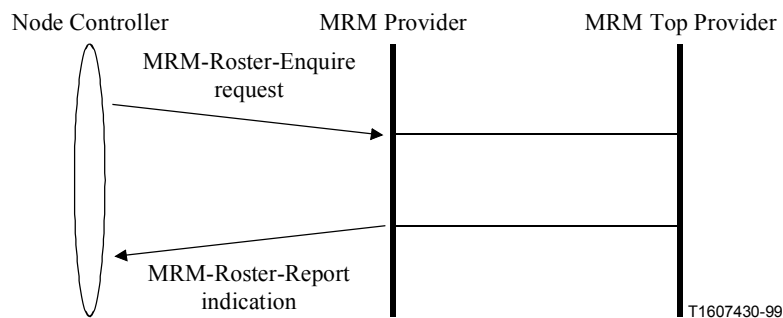


Figure 9/T.137 – MRM-Roster-Enquire – Sequence of primitives

Table 9/T.137 – MRM-Roster-Enquire – Types of primitives and their parameters

Content	Request	Confirm
Scope of Request	M	–
Detail Flag	M	–
Request Configuration Information Flag	M	–
Request Status Information Flag	M	–
Request Participants Information Flag	M	–
Request Participants List Flag	M	–
Request Room Templates Flag	M	–

Scope of Request: This allows the roster enquire to specify the whole conference, a specific room, a specific group, or an individual.

Detail Flag: The MRM Rosters potentially contain a lot of information. The top provider will include information in its reports as specified in the framework. This flag allows the enquirer to request more details if they are available.

Request Configuration Information: Flag is set when information is required on the configuration for the selected scope.

Request Status Information: Flag is set when information is required on the status for the selected scope.

Request Participants Information: Flag is set when participant information is required for the selected scope.

Request Participants List: Flag is set if a list of MRM participants is required for the selected scope.

Request Room Templates: Flag is set to request room template list from the top provider.

Result: Successful, or contains the failure reason code "NotPermitted".

9.5 Meeting rooms

Meeting rooms may be predefined within a conference framework, using room constructor templates. When a framework containing a room specification is used to create a conference, the specified rooms will also be created. These rooms exist for the duration of the conference and are persistent meeting rooms. They will continue to exist even when the room is empty. Room persistence is achieved because the MRM top provider joins each of these rooms and remains there for the duration of the conference.

Alternatively, only the room constructors can be predefined as part of a framework, and then later used by conference participants to create public or private rooms.

Meeting rooms may also be created dynamically during the conference, in response to participant requirements. These dynamic rooms may be open, or have their access controlled. By default, dynamic rooms are owned by the creator and automatically terminated when the creator leaves. However, MRM allows the owner of the room to be specified, providing greater flexibility.

In all cases, it is the MRM-Room-Create primitives that are used to create rooms.

9.5.1 Description of abstract services

Following is a list and a summary for each of the primitives defined in this subclause:

- MRM-Room-Create – Used by a node controller to create a new MRM room.
- MRM-Room-Enter – Used by a node controller to enter an MRM room.
- MRM-Room-Leave – Used by a node controller to leave an MRM room.
- MRM-Room-Invite – Used by the creator of a private MRM room to invite the participation of another node.

9.5.1.1 MRM-Room-Create

Protocol link: 10.8.1

The MRM-Room-Create request primitive is used by a node controller to issue a request the creation of an MRM room, a separate meeting space within the existing conference. This room configuration may be fully specified or a room constructor template may be used, selected from a predefined framework definition.

Figure 10 shows the sequence of events when using this primitive. Table 10 shows the parameters and types of this primitives.

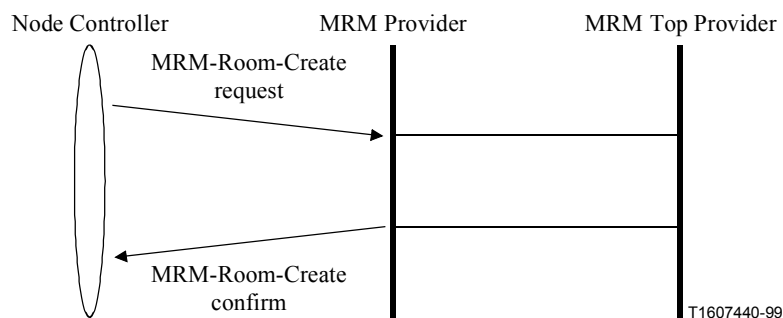


Figure 10/T.137 – MRM-Room-Create – Sequence of primitives

Table 10/T.137 – MRM-Room-Create – Types of primitives and their parameters

Content	Request	Confirm
Request Handle	M	M(=)
Room ID Handle	--	C
Room Owner Flag	M	–
Room Parent	M	–
Framework Room ID	O	–
Use Template Flag	M	–
Room Template ID	O	–
Room Mode	O	–
Room Media	O	–
Room Access Regime	O	–
Room Entry Regime	O	–
Room Exit Regime	O	–
Room Name String	O	–
Room Purpose String	O	–
Number of Participants	O	–
Number of Observers	O	–
Required Roles List	O	–
Required Services List	O	–
Room Master Key	–	M
Result	–	M

Request Handle: Locally unique number to allow the requesting node to match responses.

Room ID Handle: MRM Identifier for the room. An MCS channel ID. Supplied by the requesting MRM provider if it wishes to be the owner (this requirement is signalled to the provider by the node controller setting the ownership flag). In the absence of a Room ID being provided, the MRM top provider becomes the owner and provides a Room ID in the response.

Room Owner Flag: Flag indicates if the requester wishes to own the room; if it is not set, then it is the MRM Top Provider that assumes ownership.

Room Parent: Room ID handle for the parent room to which the new room should attach.

Framework Room ID: This is a room identifier that has been pre-assigned to the room outside of the protocol through the framework mechanism. This parameter should therefore only be used by the Top MRM Provider in response to a framework being used to create a conference.

Use Template Flag: Flag is set to TRUE if room specification is to be based upon framework constructor template.

Room Template ID: Identifiers of a predefined template to be used to create the room; if supplied, the only other parameters that will be accepted are the numbers and strings. Other parameters should be left blank.

Room Mode: Permanent, Persistent, Dynamic.

Room Media: Audio/Video/other.

Room Access Regime: Open, Controlled, Private.

Room Entry Regime: Choice of: Direct, Chair mediated, By invitation, via lobby.

Room Exit Regime: Return to: Previous/Lobby/Exit conference.

Room Hierarchy Support: Integer between 1 and MaxTreeDepth.

Number of Participants: Integer between 0 and max participants.

Number of Observers: Integer between 0 and max participants.

Room Name String: Optional name string for room.

Room Purpose String: Optional room purpose description string.

Required Roles List: Optional list of required roles.

Required Services List: Optional list of required services.

Room Master Key: Key returned by MRM top provider when a room is created; it will give holder access to this room irrespective of entry and access regimes in place.

Result: Indicates whether the request was successful or if it failed one of the following reason codes: Not authorized, Prohibited by framework, Invalid template, Room depth exceeded, Max rooms exceeded, Invalid parent ID.

9.5.1.2 MRM-Room-Enter

Protocol link: 10.8.2

The MRM-Room-Enter request primitive is used by a node controller to move itself from a lobby or a meeting room into another MRM meeting room. A room may have an entry regime set-up that adds an extra stage to the room entry procedure. The node responsible for policing the entry regime is termed the "Door keeper" and is the arbiter of room entry requests. The "Door Keeper" role is by default assigned to the operator; however, it may be reassigned through the role mechanism. If the MRM-Room-Enter request fails, then the participant will remain in its current location.

Figures 11 and 12 show the sequence of events when using this primitive.

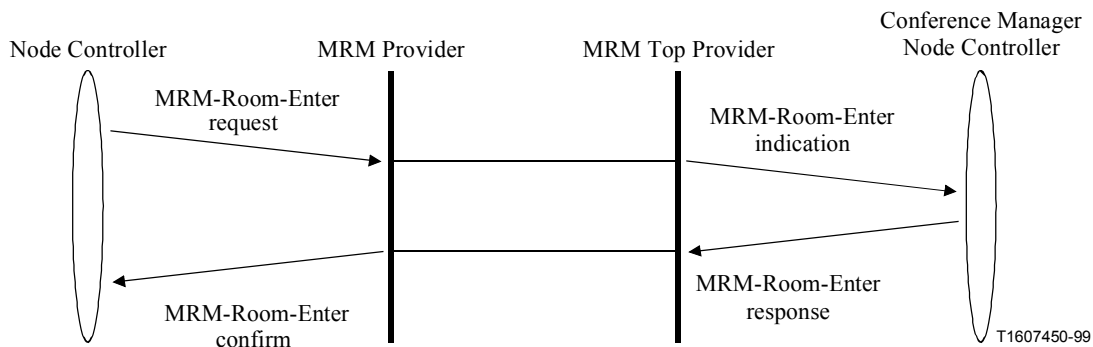


Figure 11/T.137 – MRM-Room-Enter – Sequence of primitives

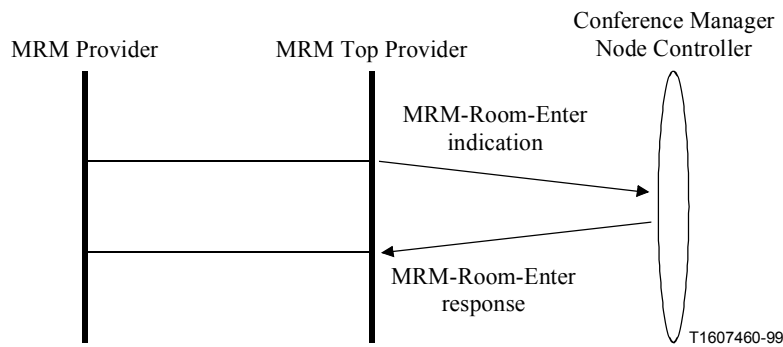


Figure 12/T.137 – MRM-Room-Enter: In response to MRM Top Provider command – Sequence of primitives

Where a role holder such as the chair is required to authorize room access, there is an additional protocol step required to get permission from the arbiter to enter the room. Once the arbiter's permission is obtained, the room entry procedure continues in the normal way. (See Figure 13.)

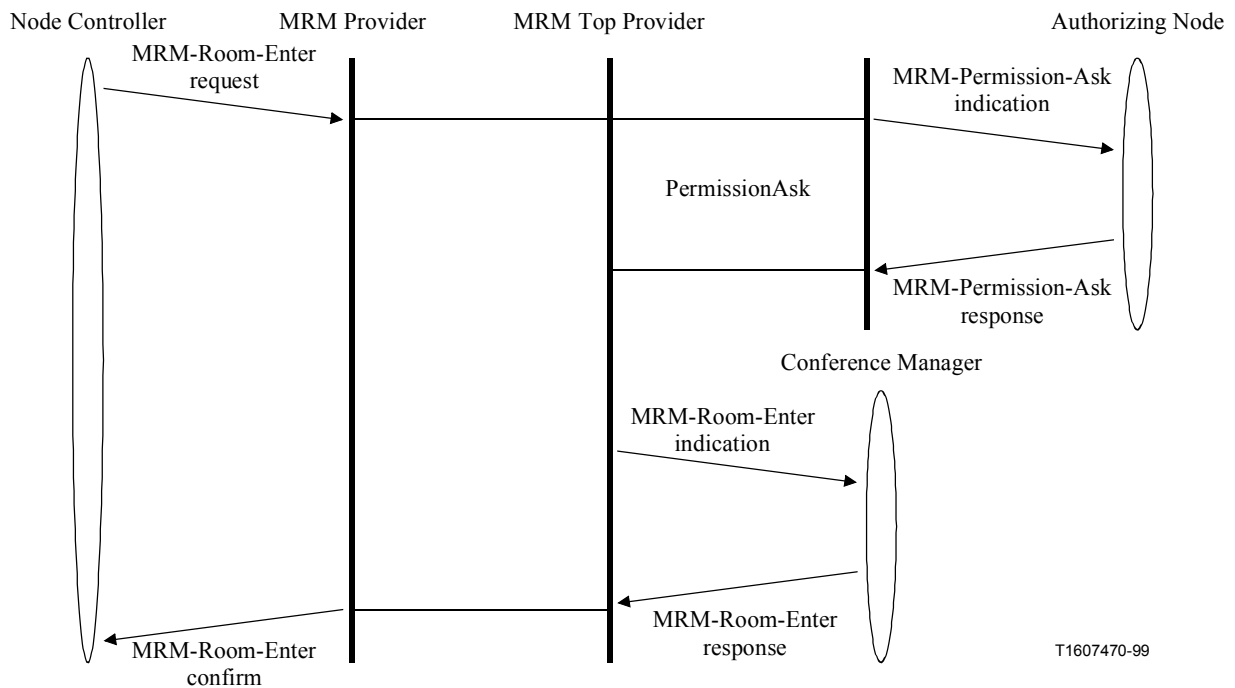


Figure 13/T.137 – MRM-Room-Enter: With an arbiter to authorize entry – Additional sequence of primitives

Table 11 shows the parameters and types of this primitive.

Table 11/T.137 – MRM-Room-Enter – Types of primitives and their parameters

Content	Request	Indication	Response	Confirm
Request Handle	M	–	–	M(=)
Requesting Node ID		C		
Enter Room ID	M	M(=)	M	M(=)
Current Room ID	M	M(=)		
Type of Participation	C	C(=)		
Terminate/Suspend Flag	M	M(=)		
Goto Room ID	–	–	–	O
Channel to Join	–	–	–	O
Room Access Parameter	C	C(=)		
Role Access Parameter Key	C	C(=)		
Reason for Entry	O	O(=)	O	O(=)
Result		M	C	M

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node ID: Required when there is an arbiter for room enter requests.

Enter Room ID: Room to be entered.

Current Room ID: Room ID for current location of node.

Type of Participation: If a Room ID is specified, the node entering the room may indicate if it wishes to participate as an observer or as a participant. In the absence of this parameter, this is determined automatically, based upon the node's participation in GCC, as a conventional node, or as an anonymous node. If provided, this parameter will be verified, and nodes that are anonymous will not be allowed to take part as MRM participants within a room.

Terminate/Suspend Flag: This flag tells the protocol if the user is temporarily leaving the previous room and wishes to maintain state and data applications for a later return.

Goto Room ID: When the result parameter tells the requesting node to go to another room, e.g. the reception room, this parameter contains the Room ID for the nominated room.

Channel to Join: Provided by the top provider to tell the entrant which channel to join, used in the case where the entrant provides a key instead of an Enter Room ID.

Room Access Parameter: For controlled access rooms, an entry key is needed. When this is provided inband, it is contained in this parameter: *Password*.

Role Access Parameter Key: Key provided out of band that will allow node to be automatically assigned a role.

Reason for Entry: Optional text string which will be added to the room roster database.

Result: Indicates whether the request was successful. It contains one of the following results: Successful, Private room, Goto specified room, Invalid room ID, Not authorized.

9.5.1.3 MRM-Room-Invite

Protocol link: 10.8.3

The MRM-Room-Invite request primitive enables the creator of a private room to invite nodes to participate in that room. Such nodes may then join the designated room by using the MRM-Room-Enter primitive.

Figure 14 shows the sequence of events when using this primitive. Table 12 shows the parameters and types of this primitives.

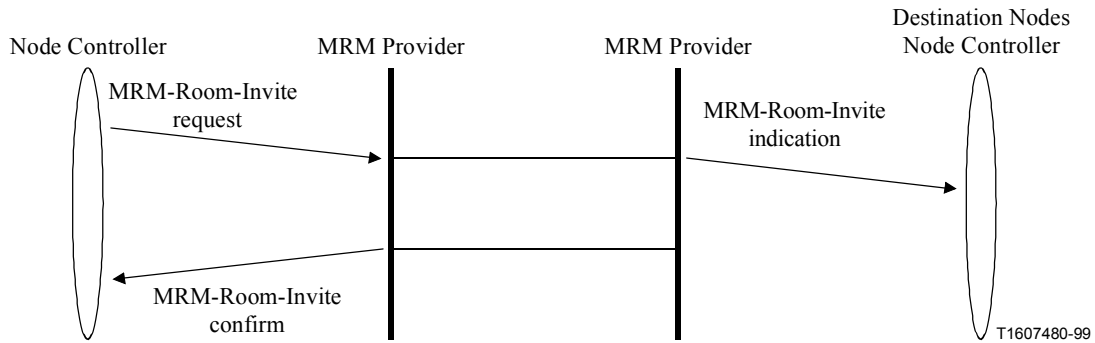


Figure 14/T.137 – MRM-Room-Invite – Sequence of primitives

Table 12/T.137 – MRM-Room-Invite – Types of primitives and their parameters

Content	Request	Indication	Confirm
Request Handle	M	–	M(=)
Room ID	M	M(=)	
Room Entry Key	M	M	
Invited Node List	M	–	
Invitor's MRM ID	–	M	
Reason String	O	O(=)	
Result		O(=)	M

Request Handle: Locally unique number to allow the requesting node to match responses.

Room ID: The MRM room to which nodes are being invited.

NOTE – RoomID = **MRM-Room-Communication-Channel**.

Room Entry Key: Password or other form of access key needed to enter the room.

Invited Node List: List of MRM User IDs to be invited.

Inviter's MRM ID: MRM User ID of the node which initiated the MRM-Session-Invite request.

Reason String: This parameter can be used to provide a human-readable string with reason for invitation in the request.

Result: Successful, Invalid Room ID, Not room owner.

9.6 MRM groups

An MRM group is an association of nodes participating *within* an MRM room. This association can be used to allow the group members to be treated as a single entity for the purpose of control or indications, for example allowing all group members to be muted or mixed. Group membership is a type of room role that room participants can take on or be assigned. MRM allows groups to be pre-assigned by means of the conference framework and it also allows groups to be defined by the convenor, or by authorized participants when the room is active. Further to this, MRM supports the advertising of groups, allowing them to be joined by participants. A group may have an owner defined. By default this will be the creator; however, it may be assigned to another node.

9.6.1 Specifying groups in the framework

If the conference convenor wishes to pre-define groups for use within a conference, then this must be done as part of the conference framework. Usually, group definitions are done within the context of a specific room. However, the group scope flag may be set to conference instead of room, allowing a group to have membership across the entire MRM conference.

Room and group requirements are realized when a framework is loaded into a created conference. Like rooms, MRM groups are assigned a unique identifier that also provides a communication channel for that group. When a group is specified in the conference framework, it is assigned a key, that is the means by which nodes can enter and be added to the group automatically. Nodes with pre-assigned roles or groups will only be able to identify or specify the group by its framework identifier. So, once the conference is started, this identifier must be associated by the protocol with the assigned MRM identifier for the group.

The convenor may pre-define a group by assigning it a Framework Group ID; the convenor may optionally specify a group name and purpose that are both human-readable strings, providing a means for participants to identify the group. If the convenor wishes to define an ordering of group members, for example to provide an order of presenters, then an access key must be specified for each member. Each key must be unique and the order of specification will represent the order within the group.

9.6.2 Specifying groups within an active room

If the convenor or an authorized participant wishes to define a group in an active room, then the MRM-Group-Create request primitive is used. When a group is created, an MCS channel is assigned to the group and used as the identifier and communication channel for that group. The group may, if required, have a human-readable name and/or a purpose description assigned to it. Any names and description are made available to participants as appropriate through the MRM roster mechanism. Group participation may be expressed as a list of MRM User IDs taken from the room roster and/or an access key which will be used to validated nodes requesting to join. Group information is always public but the ability to join a group may be dependent on having a valid access key. A group member leaving a room will also be removed from a group if the group's scope is limited to just that room.

9.6.3 Description of abstract services

Following is a list and a summary for each of the primitives defined in this subclause:

- MRM-Group-Create – Used by a node controller to associate one or more nodes within an MRM conference for the purpose of control and management.
- MRM-Group-Disband – Used to remove an existing group definition.
- MRM-Group-Membership-Specify – Used to specify which nodes shall become part of the MRM group.

9.6.3.1 Creating a group

Figure 15 shows the sequence of events when using this primitive. Table 13 shows the parameters and types of this primitives.

Protocol link: 10.8.4

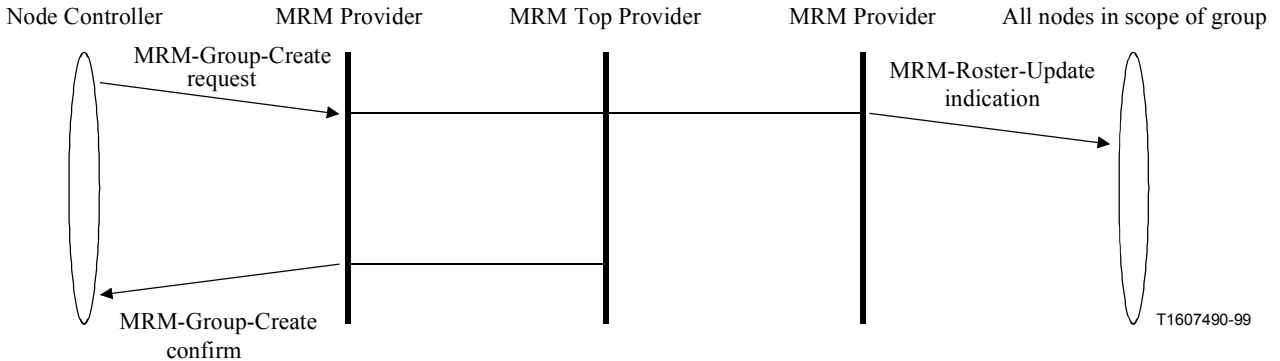


Figure 15/T.137 – MRM-Group-Create – Sequence of primitives

Table 13/T.137 – MRM-Group-Create – Types of primitives and their parameters

Content	Request	Confirm
Request Handle	M	M(=)
Group Owner's MRM User ID	O	–
Group Scope Flag	M	–
Room ID Handle	C	–
Group ID	–	M
Group Name	O	–
Group Function	O	–
Ordered Group Flag	M	–
Group Member List	O	–
Result	–	M

Request Handle: Locally unique number to allow the requesting node to match responses.

Group Owner's MRM User ID: The creator of the group specifies which node will own the group.

Group Scope Flag: Flag specifies the scope of the group.

NOTE – This will normally be a room. Setting the flag to TRUE makes the scope conference wide.

Room ID Handle: The Room Identification Handle which the group is to be associated with.

Group ID: The Group ID is assigned by the top MRM provider and is guaranteed unique within that conference.

Group Name: Optional name string for the group.

Group Function: Optional description string for the group.

Ordered Group Flag: Flag specifies that the group is to be an ordered group.

Group Member List: Optional list of group participants.

Result: Result of the Group-Create request.

9.6.3.2 Disbanding a group

Figure 16 shows the sequence of events when using this primitive. Table 14 shows the parameters and types of this primitives.

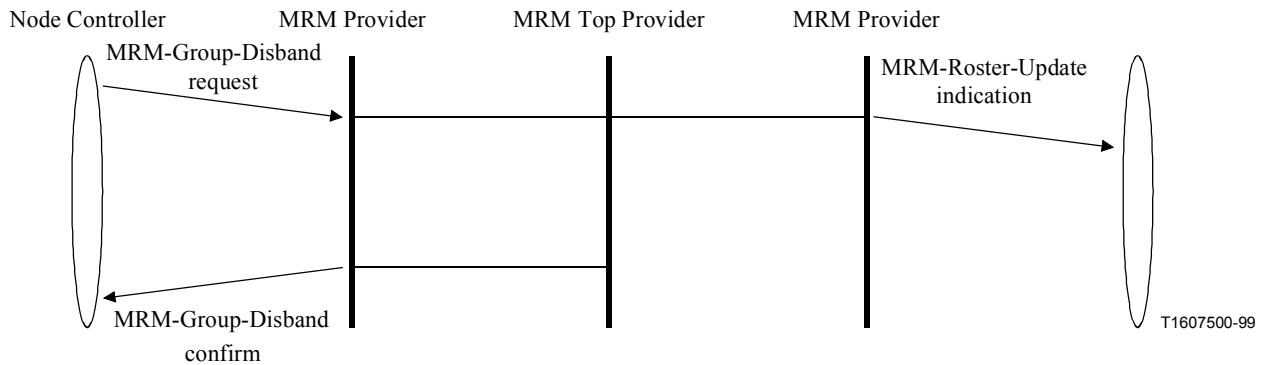


Figure 16/T.137 – MRM-Group-Disband – Sequence of primitives

Table 14/T.137 – MRM-Group-Disband – Types of primitives and their parameters

Content	Request	Confirm
Request Handle	M	M(=)
Group ID Handle	M	M(=)
Result	–	M

Request Handle: Locally unique number to allow the requesting node to match responses.

Group ID Handle: MRM User ID for the group.

Result: Result of disband request returned by top provider – Successful, Not group owner, Invalid group ID.

9.6.3.3 Assigning group membership

Figure 17 shows the sequence of events when using this primitive. Table 15 shows the parameters and types of this primitives.

Protocol link: 10.8.4

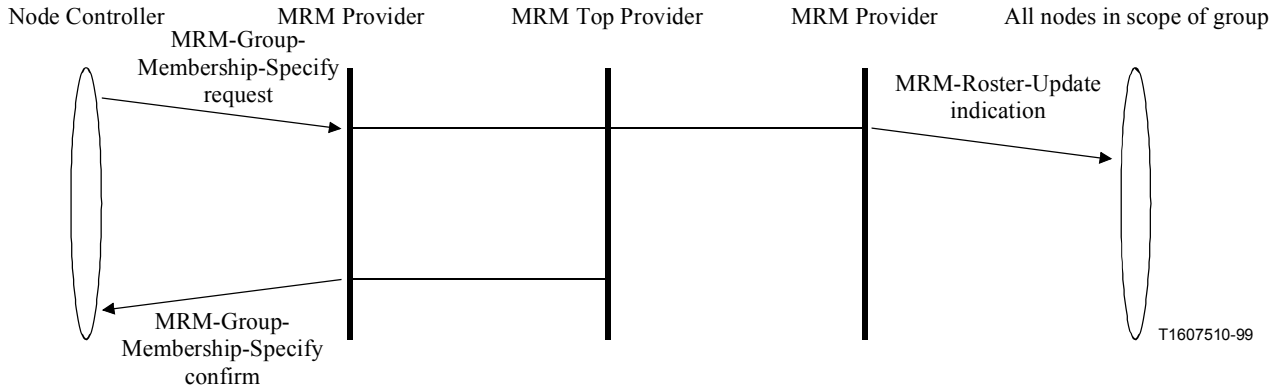


Figure 17/T.137 – MRM-Group-Membership-Specify – Sequence of primitives

Table 15/T.137 – MRM-Group-Membership-Specify – Types of primitives and their parameters

Content	Request	Confirm
Request Handle	M	M(=)
Group Identifier	M	–
Add Member List	O	–
Remove Member List	O	–
Specify Complete Member List	O	–
Result	–	M

Request Handle: Locally unique number to allow the requesting node to match responses.

Group Identifier: User ID obtained by top provider and used to represent group.

Add Member List: List of MRM User IDs from the room or conference roster.

Remove Member List: List of MRM User IDs from the room or conference roster to be removed from group.

Specify Complete Member List: Complete group membership list of MRM User IDs from the room or conference roster.

Result: Result of group membership specification request. Either Success, or List contain successful assignments, or Not authorized to add to group.

9.6.3.4 Group leader

When a group has been defined, it is possible to use the role primitives to define a group leader who can act as spokesperson or arbiter for the group. This is done by specifying a group identifier in the MRM-Room-Role-Assign request. This role only has privileges over members of the same group. A group leader cannot be compared with other roles, which have jurisdiction across the whole room.

When the room in which the group is defined has a chair, the group leader sits between the other group members and the chair and can arbitrate between competing requests to speak, to be seen, or to use the data applications.

9.7 T.120 sessions

MRM provides the potential to segregate a GCC conference into a number of MRM rooms. To fully realize the potential of the MRM model, T.120 application sessions should have the ability to have their scope limited to only the participants within a particular MRM room. MRM provides the session association mechanism to allow data application sessions to be associated with an MRM room, thus signalling the requirement for a reduced scope for that data session.

9.7.1 Description of abstract services

- MRM-Session-Associate – The means to indicate a data application session is to be associated with an MRM room.

9.7.1.1 MRM-Session-Associate

Protocol Link: 10.9

The MRM-Session-Associate request primitive is used by a node controller to associate an MRM session with one or more data sessions. A successful request supersedes any previous associations for a particular data application. The primitive may thus also be used to terminate associations. By default, all data sessions are implicitly associated with the MRM Terminal Standard Default Room. Nodes can determine which data sessions are associated with a particular MRM room from the MRM room roster for that room. The data sessions are referenced using their registry key as defined in Recommendation ITU-T T.120.

Figure 18 shows the sequence of events when using this primitive. Table 16 shows the parameters and types of this primitives.

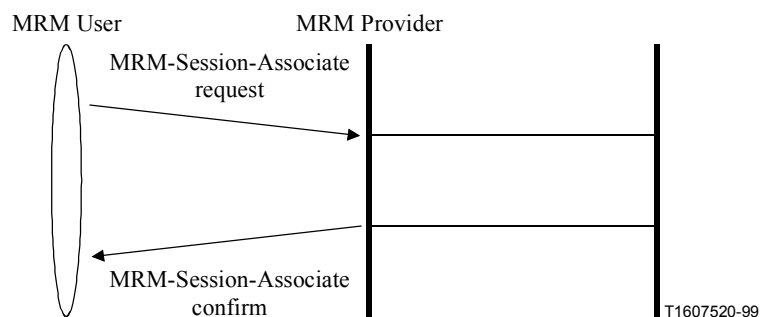


Figure 18/T.137 – MRM-Session-Associate – Sequence of primitives

Table 16/T.137 – MRM-Session-Associate – Types of primitives and their parameters

Content	Request	Confirm
Request Handle	M	M(=)
Room ID	M	–
Associated Sessions	M	
Result		M

Request Handle: Locally unique number to allow the requesting node to match responses.

Room ID: The Room ID for the MRM room to which the following application protocol sessions are to be associated.

Associated Sessions: A list of application keys and Session IDs of data application protocol sessions with which the specified MRM session is to be associated.

Result: Indicates whether the request was successful. It contains one of the following results: Successful, Invalid session ID, Not authorized node.

9.8 MRM room roles

An MRM conference supports participation of both conventional nodes and anonymous nodes. In the MRM protocol these node types define the level of participation in the MRM conference and are effectively conference roles. The anonymous role is used by a node joining the MRM conference as an observer, while the conventional role is used by a fully functional participant in the conference. Only conventional nodes are permitted to take on additional conference or room roles.

MRM provides support for a convenor role, which is effectively a default administration role allowing extensive access to creation and management tools. Users are free if they wish to define their own administrator roles by assigning the required role attributes to a new role definition. MRM also provides support for a standardized room chair role. This provides a capability for management of the proceedings within a room.

The chair role for the default room may be aligned with the GCC conductor role (defined in Recommendation T.124) but unlike the GCC roles which are conference wide, these roles are only valid within the context of an MRM room. The MRM convenor role can be regarded as an extension of the GCC convenor's capabilities to support the requirements of MRM. The GCC/MRM convenor is assigned at conference creation time and is not a role that may be taken on within the conference.

The chair and the user-defined roles utilize the MRM-Room-Role primitives, by using the role field to specify which role is being addressed. Non-standard identifiers are used to identify user-defined roles.

The MRM conference framework is typically used to put bounds upon resource requirements for the conference. It can also be used to impose a required usage model and management regime. For events where role holders are known in advance, the framework will not only allow the specific role to be pre-defined within a room, but also allow access keys to be associated with given roles. These keys are passed out of band to the defined role-holder. When a participant with a pre-defined role and an access key enters a room and makes a role assignment request, the request is validated by the MRM top provider against the framework and the role is assigned if the key is valid.

The pre-assignment of roles and access keys using the framework allows the establishment of potentially complex conference scenarios to be automated and minimizes the complexity for the participant.

The node controller at each node will receive room role indications. If the "Synchronize Data conference" flag is set, then the Node Controller should also pass these indications on to any data application protocol sessions that are associated with that room.

MRM defines two standard roles; part of the value of these roles is that they provide a standardized behaviour for functionality that is regularly required. Table 17 shows the function attributes assigned to the standardized room roles. From the table a user-defined role can be created and function attributes assigned to it.

Table 17/T.137 – Room permission privileges – Parameters for standardized room roles and to facilitate room role creation

Roles Function permissions <i>(when role active)</i>	Convenor	Observer	Participants		
			Basic permissions	Added Chair permissions	User-defined permissions
Role permissions					
Create Role	√	x	x	√	O
Pass Role	√	x	O	√	O
Ask For Role	√	x	O	√	O
Modify Role Permissions	√	x	x	x	O
Modify Other Role Permissions	√	x	x	x	O
Chair Authorizes Role request	√	x	C	x	O
Room management					
Create/Modify/Destroy child rooms	√	x	O	√	O
Child Rooms Inherit Parent Roles	√	x	O	√	O
Authorize Room Entry (Door Keeper)	√	x	x	√	O
Modify Room Parameters	√	x	x	√	O
Create/Modify/Destroy groups	√	x	O	√	O
Change Group Ordering	√	x	x	√	O
Start/Stop floor Control	√	x	x	√	O
Authorize Room transactions	–	–	–	O	O
Request Floor	–	O	O	–	O
Operate Floor Control	√	x	x	√	O
Invite Participants	√	x	O	√	O
Eject participants from Conference	√	x	O	√	O
Data application control					
View Data Apps	O	O	O	√	O
Control Data Apps	O	x	x	√	O
Exclusive Control of Data App(s)	O	x	x	O	O
Launch Data Applications	O	O	O	√	O
View Specific Application	O	O	O	√	O
Control Specific Application	O	x	O	√	O
Control Specific Device	O	x	O	√	O
Service management					
Launch Media Services	C	x	x	√	O
Activate Media Services	C	x	x	√	O
Audio					
Control Audio Mixes	O	x	x	√	O
Control Audio Devices	O	x	x	√	O
Invite audio Only participants	O	x	O	√	O
Mutable By others	x	√	√	x	O

Table 17/T.137 – Room permission privileges – Parameters for standardized room roles and to facilitate room role creation (*concluded*)

Roles Function permissions (when role active)	Convenor	Observer	Participants		
			Basic permissions	Added Chair permissions	User-defined permissions
Video					
Control Video Devices	O	x	O	√	O
View Video	O	O	O	√	O
C Conditional O Optional √ Required x Prohibited					

Table 17 shows MRM functionality classified as a number of separate permissions. It shows that the static conference role of convenor has, as conference creator, many default privileges to reconfigure and change role permissions. It also shows that an observer is prohibited from using most of the functionality available to MRM participants. MRM participants have a range of permissions available and some features restricted. The table also shows the chair, the only MRM room role defined in the protocol, and the permissions associated with that role. The final column shows that user-defined roles may be assigned with any combination of permissions as required.

9.8.1 Room role permissions – Specification

This Meeting Room Management Recommendation supports standard and user-defined specification of room-based Roles. Every role definition is effectively a unique identification label for a group of permissions and privileges to access or control additional functionality. When a room role is assigned to a room, it only has scope within that room; this is based upon the premise that a participant can only be in one room at any given time.

The role permissions list is divided into five categories: Role permissions, Room management, Data Application Management, Video Management and control, and Audio Management and control. Every person participating in an MRM conference adopts either the role of an observer or that of a participant, and these two roles are an inherent part of the MRM operational model. Effectively, everyone in an MRM conference is therefore a role holder; additional roles may be taken on or assigned using the MRM role functionality. These additional roles effectively modify the privilege list for that node. The MRM roster contains information on roles held and roles assigned.

Create/Modify Role – This permission provides the ability for the role holder to create additional roles, and arbitrarily assign privileges to those roles. It is the task of the MRM top provider to prohibit assignment of privileges the creator is not authorized to use. In most cases it is expected that roles will be pre-defined within the conference framework.

Pass Role – This permission allows the role holder permission to use MRM-Room-Role-Give functionality, allowing a role holder to give the role to another conventional node in the MRM meeting room.

Ask For Role – This permission allows the role holder permission to use MRM-Room-Role-Please functionality, allowing a role holder to request a role from another conventional node in the MRM meeting room.

Modify Other Role Holder Permissions – This permission allows a role holder to modify other role permissions at will. The MRM conference manager should also prohibit arbitrary use of this as appropriate for the operational use of the protocol.

Modify Role Holder Permitted Privileges – This permission allows a role holder to modify other role permissions only if it has that permission itself. This is the safest option to use as it does not allow dynamic enhancement of permissions within a conference.

Chair Authorizes Role request – This permission determines if the chair needs to authorize role assignment requests when a chair is assigned.

Room Management

Create child rooms – This permission allows the role holder to use the MRM room functions to create additional meeting rooms.

Make Rooms Persistent – This permission allows a role holder the capability to make an existing dynamic room persistent, but not the creation, modification, or destruction of rooms.

Create/Modify/Disband groups – This permission allows the role holder to use the MRM group functions to create, modify, and disband groups.

Modify Room Parameters – This permission allows the holder to modify parameters within a room.

Change Group Ordering – This permission allows an ordered group's ordering to be modified.

Start/Stop Floor Control – This privilege allows the role holder permission to access the floor control service.

Authorize Room Transactions – This privilege causes all transactions such as role assignments to require authorization from the holder of the privilege before proceeding. The MRM top provider generates Permission-Ask requests to obtain permission on the nodes' behalf. MRM ensures only one authorizing node can exist in any given room.

Request Floor – This permission allows the role holder to have access to the request to participate functions.

Operate Floor Control – This privilege allows the role holder permission to manage the floor control service.

Invite Participants – This permission allows the role holder the ability to invite participants to a room.

Eject Participants – Eject participants from the conference using the GCC Eject User mechanism.

Data application management

View Data Apps – This permission allows the role holder the ability to view data applications, although this is available by default for participants. The removal of this feature allows, for example, an operator to come in and assist in the conference without being able to view the data content.

Manage Data Apps – This permission allows the role holder the ability to manage data application. For most applications this may just consist of access to the controls to drive the application. MRM-aware applications may provide additional management support for management.

Exclusive Management of Data Apps – This permission allows this role holder and only this role holder access to the ability to manage data applications.

Control Specific Application – This permission allows a specific role to gain permission to access/control a specific data application which must be identified by its application ID as defined in T.120.

Control Specific Device – This permission allows a specific role to gain permission to access/control a specific device.

Audio management and control

Control Audio Mixes – This permission allows the role holder the ability to control who or what is mixed into the audio for the meeting room.

Control Audio Devices – This permission allows the role holder permission to control other audio devices available within the conference room.

Invite/Eject Audio-only participants – This permission allows the role holder to bring in audio-only participants to the conference by dialling out to them.

Mutable – This permission defines if the role holder's audio stream(s) are mutable or if they are always mixed.

Video management and control

Control Video Devices – This permission allows the role holder permission to control video devices available within the conference room.

View Video Apps – This permission allows the role holder the ability to view video applications, although this is available for participants, by default. The removal of this feature allows, for example, the ability to shield sensitive material from an operator assisting in the conference.

9.8.2 Room Roles – Description of abstract services

Following is a list of the primitives defined in this subclause and a brief summary of the function of each:

- **MRM-Room-Role-Assign** – Used by a node controller to request the assignment of a room role. When a node takes on a role, e.g. chair, a successful bid results in a roster update to the room.
- **MRM-Room-Role-Release** – Used by a node controller to release the specified role in a room. When any node releases a role, the release results in a roster update to the room.
- **MRM-Room-Role-Please** – Used by a node controller to request that a role be given to it from the current room role holder.
- **MRM-Room-Role-Give** – Used by a node controller to pass a role to a specified node.
- **MRM-Room-Role-Inquire** – Used by a node controller to determine whether the room currently has a specific active role and, if it does, what the role holder's MRM User ID is, or if no role is specified, then a list of roles present in the conference is returned.
- **MRM-Role-Permissions-Modify** – This primitive is only used by a node with role modification permissions, in order to modify role permissions.

9.8.2.1 MRM-Room-Role-Assign

Protocol link: 10.10.1

When a defined MRM role is not assigned, a participant's node controller at any node has the possibility to issue a request to take on that role, by issuing a MRM-Room-Role-Assign request primitive. The result parameter, returned in the confirm primitive, indicates if the requester has been assigned the role or not. A successful MRM-Room-Role-Assignment results in a Roster Update to all nodes that are in the room.

Figure 19 shows the sequence of events when using this primitive.

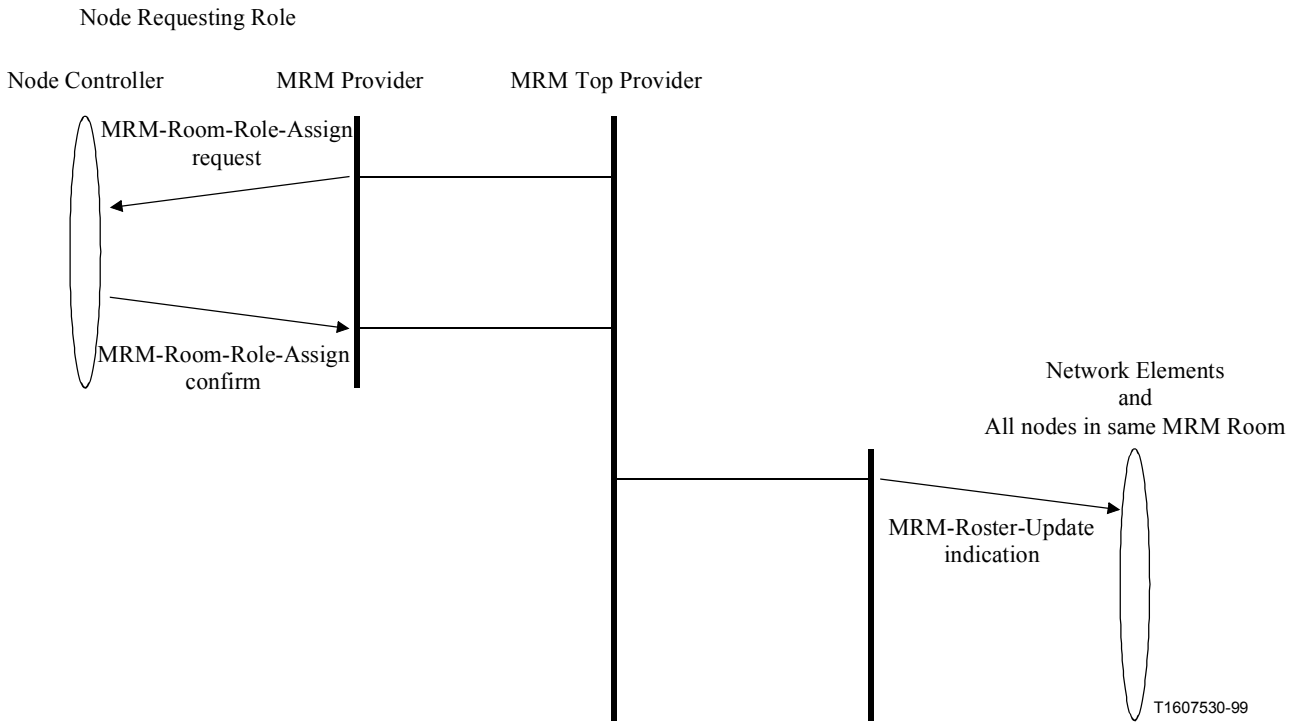


Figure 19/T.137 – MRM-Room-Role-Assign – Sequence of primitives

The assignment of a role by the MRM top provider occurs in response to a framework requirement identifying a participant (through the role key) as the role holder. (See Figure 20.)

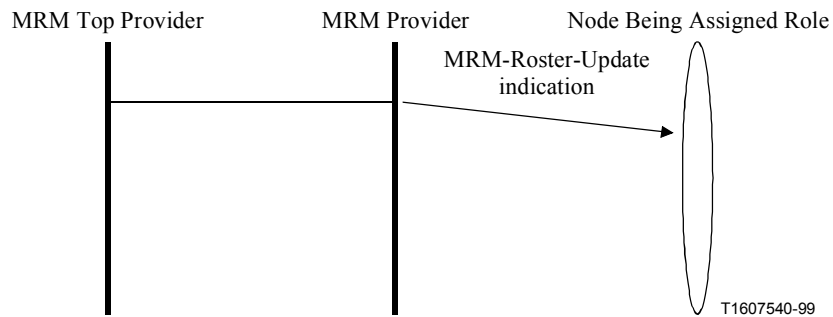


Figure 20/T.137 – MRM-Room-Role-Assign: Assignment by MRM Top Provider – Sequence of primitives

Table 18 shows the parameters and types of this primitive.

Table 18/T.137 – MRM-Room-Role-Assign – Types of primitives and their parameters

Parameter	Request	Indication	Confirm
Request Handle	M		M(=)
Requesting Node MRM User ID		M	
Proposed Role Holder MRM User ID	O	O(=)	
MRM Room Identifier	M	M(=)	
MRM Room Role Identifier	C	M(=)	
MRM Role Access Key	C	M	M
Group identifier	O	O	O
Result			M

Request Handle: Locally unique identifier to allow the requesting node to match responses.

Requesting Node MRM User ID: Identification number of the requesting node.

Proposed Role Holder MRM User ID {list}: Required if the requester is not the node requiring the role or the request is for more than one role assignment as would be the case when group membership was being defined.

MRM Room Identifier: The identifier for the MRM room – It is mandatory to provide the Room ID in the request unless an access key is provided in which case the TOP Provider can resolve the correct room and the parameter is optional.

MRM Room Role Identifier: This parameter identifies the specific room role "Chair, Presenter, Group Member". It is mandatory to provide the room role in the request unless an access key is provided in which case the top provider can still resolve the corresponding role assignment, and the parameter becomes optional.

MRM Role Access Key: This parameter provides an access key which the requester may optionally provide in a request. If provided, it is used by the MRM top provider to validate the request against the conference framework access controls.

Group Identifier: If the Group ID is specified, the assign role only has scope within that group. This allows a group leader to be defined.

Result: An indication of whether the request was accepted or rejected, and, if rejected, of the reason why. It contains one of the following possible results: Successful, Invalid room, Role not permitted, Role already assigned, Not group member.

Case of updating late joiners:

Room role assignments are recorded in the room roster and changes are notified in Roster Updates. If a new node joins after a role has been assigned, they will see the role assignment in the roster report they receive.

Case of active chair:

The MRM top provider checks if any active role has "authorize room transaction" privilege. If a node does, the MRM top provider will carry out an extra authorization process with the authorizing node in order to obtain authorization for a role assignment before proceeding.

9.8.2.2 MRM-Room-Role-Release

Protocol link: 10.10.2

When an MRM node holding an active MRM role wishes to relinquish that role, it issues an MRM-Room-Role-Release request primitive. The request is sent to the MRM top Provider which will inform all nodes in the room of the change in operational mode by means of the MRM-Roster-Update indication. The order of role assignments and role releases represents the actual order of role holder transitions. Any participant issuing a MRM-Room-Role-Release request, but not being the current role holder, shall be answered by the MRM top provider with an MRM-Room-Role-Release confirmation containing a negative result and reason description. In this situation no indications shall be sent to the other participants. The MRM top provider can initiate a role release as a result of a role holder having disconnected from the conference.

The MRM top provider will execute a role release locally and issue a RosterUpdateIndication in the case where a node is detected leaving the MRM room or the conference. This update will contain updated role information as well as updated participation information.

Figure 21 shows the sequence of events when using this primitive. Table 19 shows the parameters and types of this primitives.

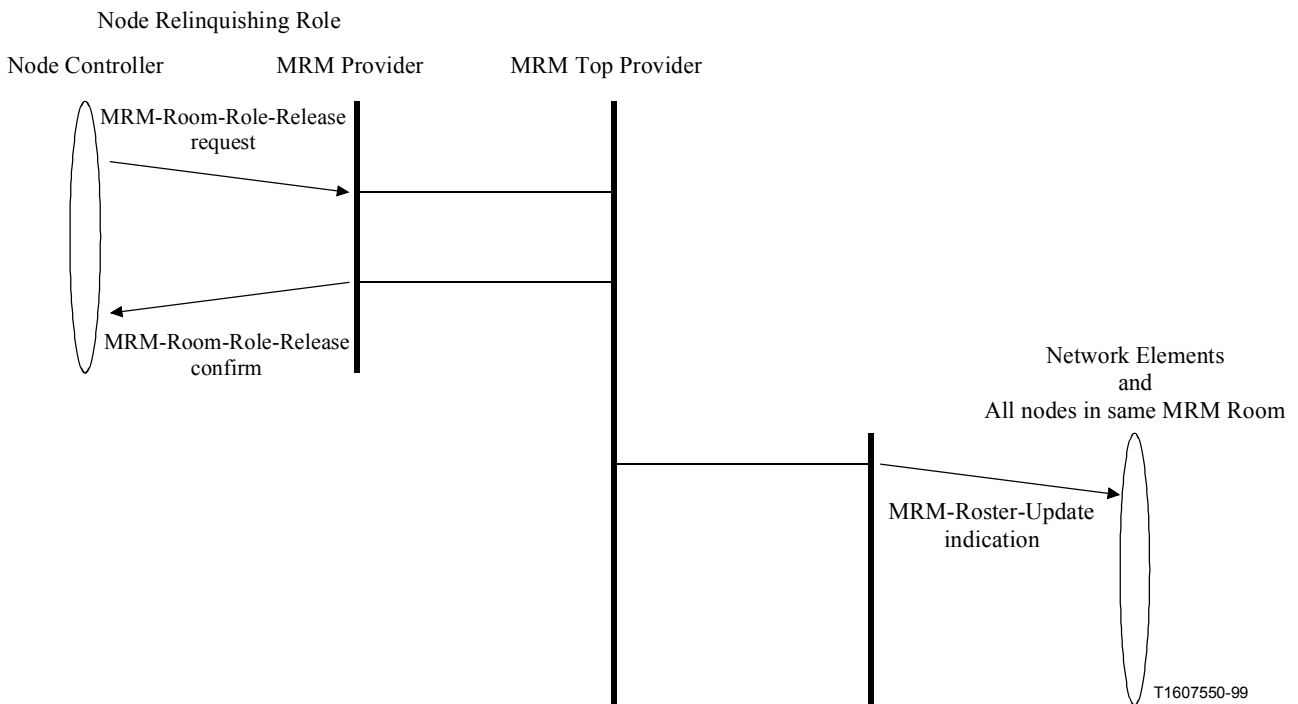


Figure 21/T.137 – MRM-Room-Role-Release (User Initiated) – Sequence of primitives

Table 19/T.137 – MRM-Room-Role-Release – Types of primitives and their parameters

Parameter	Request	Confirm
Request Handle	M	M(=RQ)
MRM Room Identifier	M	M(=)
MRM Room Role Identifier	M	M(=)
Room Role Label	C	
Group Identifier	O	
Release Reason		
Result		M

Request Handle: Locally unique identifier to allow the requesting node to match responses.

MRM Room Identifier: The identifier for the MRM room.

MRM Room Role Identifier: This parameter identifies the specific room role to be released.

Room Role Label: A single word such, as "Presenter", used to identify a user-defined role.

Group Identifier: The group identifier is an optional parameter; it allows a role within a group to be identified.

Release Reason: The reason parameter can provide additional information within indication messages concerning the release, e.g. presenter initiated, or system initiated.

Result: An indication of whether the request was accepted or rejected, and, if rejected of the reason why. It contains one of the following possible results: Successful, Invalid room, Not role holder, Invalid group.

9.8.2.3 MRM-Room-Role-Please

Protocol link: 10.10.3

The primitive MRM-Room-Role-Please request may be issued by a node controller to ask the current role holder to give the role to the requesting node. The MRM-Room-Role-Please indication is forwarded to the current role holder. The current role holder may then choose to give the role to the requester by using the MRM-Room-Role-Give primitive. The confirm portion of this primitive is only local confirmation that the request was accepted by the local MRM provider. No confirmation from the role holder is given directly.

NOTE – This primitive is not applicable to groups, only to individual roles.

Figure 22 shows the sequence of events when using this primitive. Table 20 shows the parameters and types of this primitives.

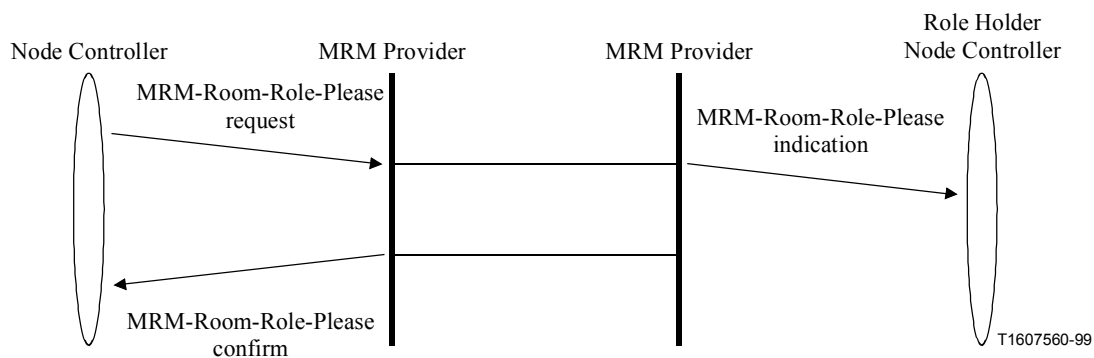


Figure 22/T.137 – MRM-Room-Role-Please – Sequence of primitives

Table 20/T.137 – MRM-Room-Role-Please – Types of primitives and their parameters

Parameter	Request	Indication	Confirm
Request Handle	M	–	M(=)
Requesting Node MRM User ID		M(=)	
MRM Room Identifier	M	M(=)	
MRM Room Role Identifier	M	M(=)	
Group Identifier	O	O(=)	
Result			M(=)

Request Handle: Locally Unique identifier to allow the requesting node to match responses.

Requesting Node MRM User ID: The MRM User ID of the requesting node.

MRM Room Identifier: The MRM Room Identifier the room request relates to.

MRM Room Role Identifier: Chair, Presenter.

Result: An indication of whether the request was accepted or rejected, and, if rejected, of the reason why. It contains one of the following possible results: Successful, Not role holder.

9.8.2.4 MRM-Room-Role-Give

Protocol link: 10.10.4

The primitive MRM-Room-Role-Give request may be issued by a node controller at the role holder's node, to transfer a role to another node. If the role is not accepted by the intended recipient, the role continues to be held by the current holder.

Figure 23 shows the sequence of events when using this primitive. Table 21 shows the parameters and types of this primitives.

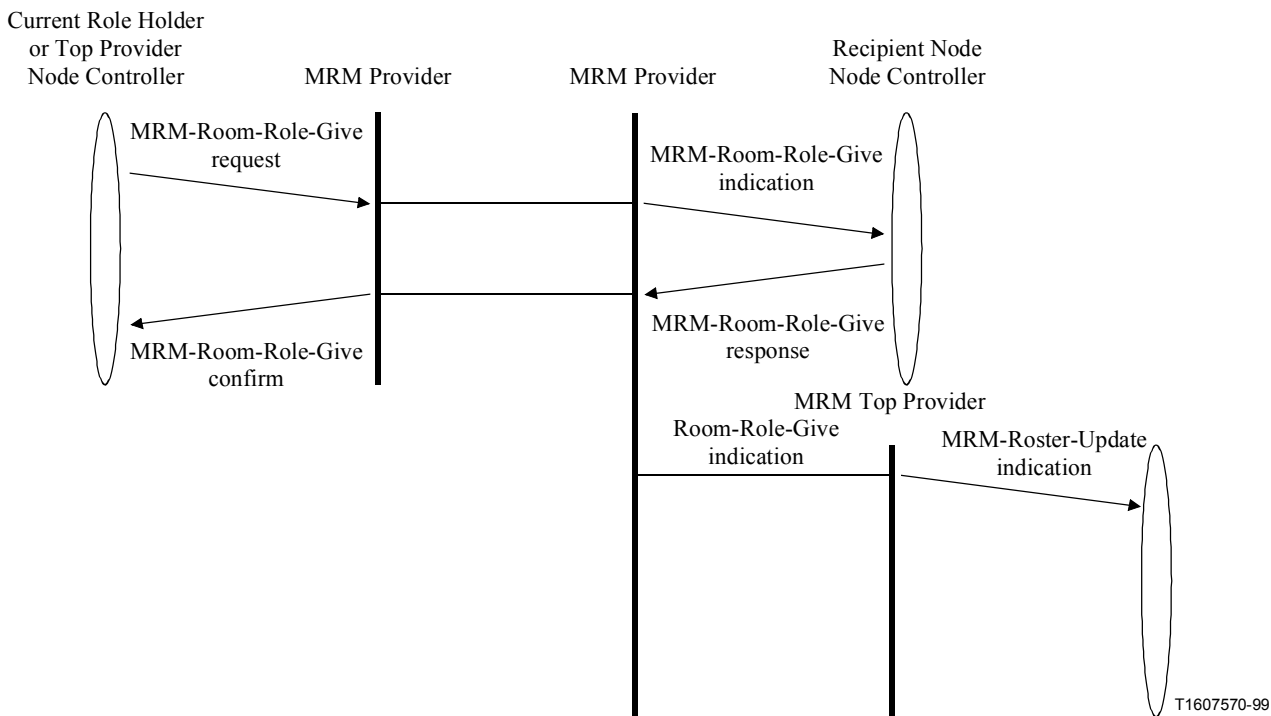


Figure 23/T.137 – MRM-Room-Role-Give – Sequence of primitives

Table 21/T.137 – MRM-Room-Role-Give – Types of primitives and their parameters

Parameter	Request	Indication	Response	Confirm
Request Handle	M			M(=RQ)
Role Holder's MRM User ID		M	M(=)	
Recipient Node	M	M(=)	M(=)	
MRM Room Identifier	M	M(=)	M(=IN)	M(=RQ)
MRM Room Role Identifier	M	M(=)	M(=)	M(=)
Group Identifier	O	O(=)	O(=)	O(=)
Result			M	M(=)

Request Handle: Locally unique identifier to allow the requesting node to match responses.

Role Holder's MRM User ID: Identification number of the role holder's terminal.

Recipient Node: The MRM User ID of the node to which the role is being transferred.

Group Identifier: The group identifier is an optional parameter; it allows a group to be defined by participants. This parameter is ignored in cases where it is not applicable, e.g. if the role is that of chair – as only one chair is permitted in a room. In the case of the presenter role, the default presenter group panel {0} will be automatically assumed if no group is specified.

Result: An indication of whether the request was accepted or rejected and, if rejected, of the reason why. It contains one of the following possible results: Successful, Invalid room, Not role holder, Give not accepted.

9.8.2.5 MRM-Room-Role-Enquire

The primitive MRM-Room-Role-Enquire request may be issued at any time by either a node controller or application protocol entity in order to find out if a specific role is active or not, and, if so, which node is the role holder. If the information is not stored locally, the MRM provider may need to retrieve it from the MRM top provider.

Figure 24 shows the sequence of events when using this primitive. Table 22 shows the parameters and types of this primitives.

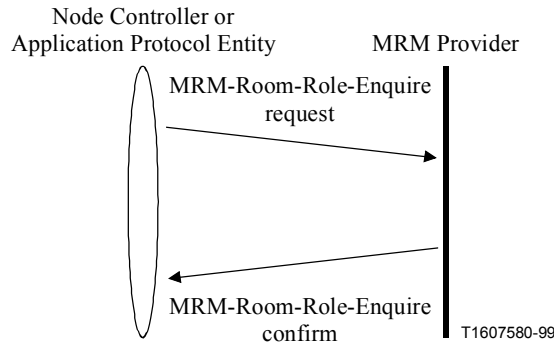


Figure 24/T.137 – MRM-Room-Role-Enquire – Sequence of primitives

Table 22/T.137 – MRM-Room-Role-Enquire – Types of primitives and their parameters

Parameter	Request	Confirm
MRM Room Identifier	M	M
MRM Room Role Identifier	M	M
Role Holder Node (MRM User ID)		C
Group Identifier	O	O(=)
Result		M

MRM Room Identifier: MRM ID of the room.

MRM Room Role Identifier: The role identifier for the role.

Role Holder User ID: The MRM User ID of the node that is currently role holder. Not present if the role is currently unassigned.

Group Identifier: The group identifier is a numeric identifier that allows groups to be distinguished. It is an optional parameter and is ignored in cases where it is not applicable – e.g. if the role is chair – as only one chair is permitted in a room. In the case of the presenter role, the default presenter group Panel {0} will be automatically assumed if no group is specified.

Result: An indication of whether the request was accepted or rejected, and, if rejected, of the reason why. It contains one of the following possible results: Successful, Invalid room ID, Invalid Role ID, Invalid Group ID.

9.8.2.6 MRM-Role-Permissions-Modify

Protocol link: 10.10.5

An MRM role holder with appropriate permissions is able to modify the role permissions of other role holders in the same room. However, the node modifying permissions is only able to modify permissions that it itself has privileges for. This means only the convenor has access to all role permissions once the conference has started. Participants can only enhance their permissions set by taking on an available pre-defined role.

Figure 25 shows the sequence of events when using this primitive. Table 23 shows the parameters and types of this primitives.

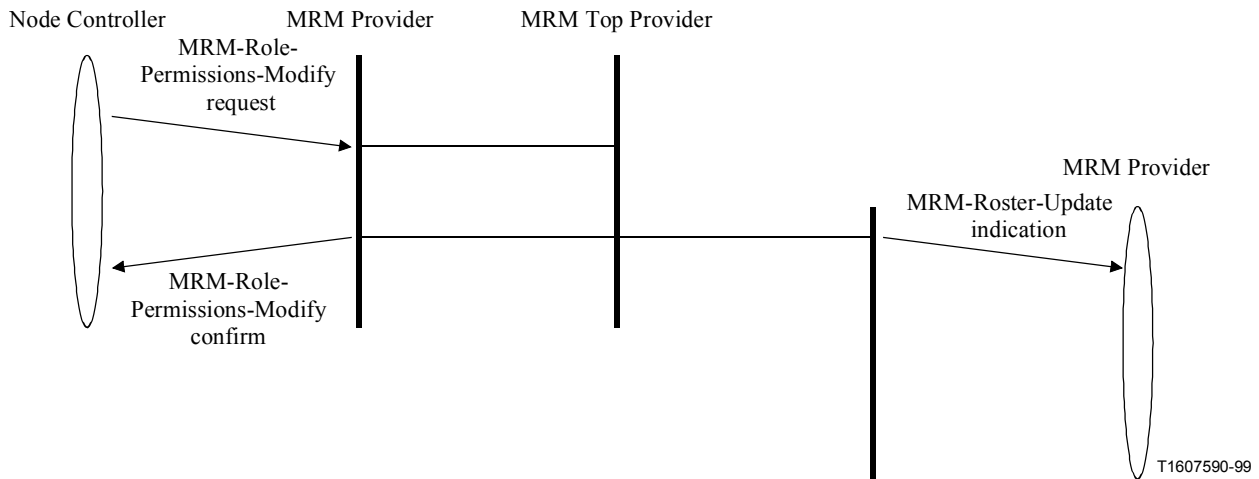


Figure 25/T.137 – MRM-Role-Permissions-Modify – Sequence of primitives

Table 23/T.137 – MRM-Role-Permissions-Modify – Types of primitives and their parameters

Parameter	Request	Confirm
Request Handle	M	M(=)
Role Holder MRM User ID	M	
Suspend/Resume Flag	M	
Permissions List	C	C
Result		M(=)

Request Handle: Locally unique identifier to allow the requesting node to match responses.

Role Holder MRM User ID: The User ID of the role holder whose permissions are to be modified.

Suspend/Resume Flag: This flag allows a role holder to suspend or resume a role it holds, allowing a role holder to continue to hold a role but to stop any actions that result from the privileges that it holds, until the role is resumed again.

Permissions List: A list of privilege flags: Create Role, Pass Role, Ask for Role, Modify Role Permissions, Modify Other Role Holder Permissions, Chair Authorizes Role request, Create/Mod/Destroy child rooms, Child Rooms Inherit Parent Roles, Create/Modify/Destroy groups, Change Group Ordering, Start/Stop floor Control, Request Floor, Modify room parameters, Operate

Floor Control, Authorize room transactions, Invite/Eject participants, View Data Apps, Control Data Apps, Control Devices, Exclusive Control of Data App(s), View Specific Application, Control Specific Application, Control Audio Mixes, Control Audio Devices, Invite audio Only participants, Mutable, Control Video Devices, View Video.

Result: An indication of whether the request was accepted or rejected, and, if rejected, of the reason why. It contains one of following possible results: Successful, Not authorized, Invalid role holder ID.

9.8.3 The chair role

The chair role operates within the context of the MRM conference room in which that role is assigned. A chair role may be pre-assigned by a convenor as part of the framework, the convenor may also assign a chair access key for the room. A particular participant who has been pre-assigned a role such as the chair can be provided with the access key (out of band) corresponding to the access key for that role in the framework. When the pre-defined chair joins the conference, the access key is passed to the conference manager which will validate the key against the framework and automatically generate an appropriate role assignment resulting in the designated node being assigned the requested role.

If the chair role is not prohibited by the current conference framework, then MRM participants may request and compete for the role as required and, if successful, act as chair for the room. If the chair is defined in the framework with an access key only the requester who provides the correct access key is assigned the role. The framework does not prohibit the role being asked for by participants or being given by the chair role holder.

An active chair automatically becomes the arbiter for the floor control service and, consequently, the floor control service requires a chair in order to operate. Only one chair is permitted within a room; therefore a conference containing multiple rooms may have multiple chair roles active.

9.8.3.1 Chair assignment

In a room without a chair, the node controller at any participating node has the possibility to issue a request to become the chairperson, by issuing an MRM-Room-Role-Assign request primitive with the identifier set to chair. The receipt of the associated MRM-Room-Role-Assign confirm primitive indicates whether the requester has become the chair or not, depending on the result parameter within the primitive. A successful MRM-Room-Role-Assign request shall be accompanied by an MRM-Roster-Update-Indication, sent to all node controllers at all nodes in the room, signalling that the room has a chair, and giving information about the identity of the chair. See Table 18 and Figure 19.

9.8.3.2 Releasing the chair role

In order to relinquish the chair role, the node controller at the chair node issues the MRM-Room-Role-Release request primitive. As soon as the Chair has requested the release of chair, all node controllers in the room are informed of the change in operational mode by means of the MRM-Roster-Update indication. Any participant issuing an MRM-Room-Role-Release, but not being the current chair, shall be answered by MRM with an MRM-Room-Role-Release confirmation containing a negative result and reason description. In this situation no indications shall be sent to the other participants. Apart from this user-initiated transition to non-conducted mode, the release of chair role can also be initiated by MRM itself, for instance because the chair leaves the room or is disconnected from the conference. See Table 19 and Figure 21.

9.8.3.3 Asking for the chair role

The primitive MRM-Room-Role-Please request may be issued by a node controller to ask the current chair to give the chair to the requesting node. The MRM-Room-Role-Please indication is forwarded to the current conducting node. The current chair may then choose to give the chair to the requester by using the MRM-Room-Role-Give primitive. The confirm portion of this primitive is

only local confirmation that the request was accepted by the local MRM provider. No confirmation from the Chair is given directly. See Table 20 and Figure 22.

9.8.3.4 Giving the chair role

The primitive "MRM-Room-Role-Give" request may be issued by a node controller to transfer the chair to a specific node. If the chair role is not accepted by the intended recipient, the conference chair continues to be held by the original Chair. See Table 21 and Figure 23.

9.8.3.5 Asking about the chairperson

The primitive MRM-Room-Role-Inquire request may be issued at any time by either a node controller or application protocol entity in order to find out whether the conference is chaired or not, and if so, which node is the chair, and if the requesting node has been granted conducted-mode permission. See Table 22 and Figure 24.

9.8.4 Accessing pre-defined roles

When a room participant has been pre-allocated a role within a room, this role will have been assigned an access key in order to ensure only the desired node is able to obtain that role. The convenor will have specified the role and associated key in the conference framework and this will have been passed to the intended recipient, by some means, out of band to MRM. The pre-allocated role holders must issue an MRM-Room-Role-Assign request providing the Role ID and Access Key together with a Group ID if available. This result will be sent to the top provider where it will be validated and the role assigned.

9.8.5 Asking a role holder for permission to Act

One of the role privileges is to be allowed to arbitrate actions of participants within the room or wishing to access the room. An MRM role holder such as a chair is able to act as an arbiter giving explicit permission before a room participant or observer can carry out an action.

9.9 Permission to take action (Incorporating floor control)

9.9.1 Description of abstract services

Following is a list of the primitives defined in this subclause and a brief summary of the function of each:

- MRM-Permission-Ask – This primitive is used by a node controller to ask permission from a chair, or other role-holding node.
- MRM-Permission-Relinquish – This primitive is used to return a previously requested permission, when it is no longer required.
- MRM-Permission-Grant – This primitive is used to grant a node permission to take a specified action.

The MRM permission primitives are provided to allow room participants to be able to request the use of conference services or to take specific actions from a room's chairperson or another node with authorization privileges. They are only relevant when a room has an active chair. In the absence of an active chair, or another authorizing node, nodes may act within their existing privileges. The specific meaning of requesting the floor is dependent on either the service context or the specific permission specified. For example, if audio services are active, the default action will be requesting to have audio included in the conference audio mix; however, the request has a service field that allows the floor control protocol to be used with other services. Floor control can only be used in a chaired room, and the chairperson is responsible for granting requests. Therefore, floor control can only operate when the Room has an active chairperson. When a chair is assigned, MRM-Role-Assign indications are sent to the node controllers at all nodes in the room, signalling that the room

has a chair, and giving information about the identity of the chair. When the room chair is assigned, the floor control capability becomes available and the chair can then launch the service as needed.

Once the service becomes active, all audio sources except the chair are muted at the mixer by the service using the Set-Audio-Mixer primitives. Room participants can issue floor requests which are directed to the chair, who can grant or decline the requests. Participants can release the floor when they have completed their requirements.

This service provides the mechanism that allows conference participants to make requests to speak in a chaired conference and for the chair to enable and disable participants from speaking.

The audio mixing service is aware of and utilizes three room roles that participating nodes can adopt in operating the service:

- **Chair role** – If the floor control has been launched, the chair becomes the arbiter for floor requests; if the service has not been launched, the chair may launch it, as may a convenor or convenor privileged node.

NOTE – By means of the framework or the alignment parameter in the floor control service, the conference conductor can take on the role of MRM room chair.

- **Participant role** – With respect to the floor control service, participants may be either Anonymous or Conventional nodes. All participant nodes will initially be muted at their local mixer when floor control is launched within a chaired room. Individuals or a group may request and be assigned the floor by the chair. Assignment of the floor will have the effect of having their audio stream included in the audio mix.
- **Observer role** – Observers are primarily viewers of media and events within the room; however, MRM supports the limited interaction through the floor control service. Requests to interact are made through the conference manager on the observer communication channel. The conference manager has access to the identity of observers even though they appear anonymous to other participants and do not appear in the room roster. The conference manager can generate requests for the floor on their behalf, thus allowing the possibility of interaction within the conference.
- **Audio-only role** – Audio-only participants are proxied for by the MRM top provider. The mechanism is a local matter.

It is not mandatory that all participating nodes be capable of supporting the service in order for it to be used. The chair may decide to launch the floor control service regardless: this will still result in the desired control, however, nodes not supporting the service will not be able to make floor requests in the conventional way.

When the floor control service is invoked by issuing a Service Command primitive, the initiating chair or the MRM conference framework specifies if the service is to be used for audio only or for both audio and video together. The MRM top provider will, on successful launch of the floor control service, set the Audio Mixer mode to "Floor Controlled", and the mixing service will then accept the command from the rooms' designated chair.

For audio control, the initial condition is determined by the framework. By default, all audio sources are muted at the mixer except that of the chair; video (if present) will continue in the current operational mode. Nodes other than the chair wishing to speak make a Permission-Ask request to the chair. The chair may decide to grant or refuse the request. Normally, requests will either be granted by the chair or be held in a queue in received order until the chair is ready to add a new speaker. The chair may also effectively remove any speaker by muting their audio source at the mixer.

For video control, the audio can be set to track the video regardless of previous modes in operation.

The chair initiates the control by communicating with mixer(s) via the MRM-Conference-Management-Channel. In voice-switched mode, the action of the video is linked to the output of the audio mixer and is therefore indirectly controlled as well.

9.9.1.1 MRM-Permission-Ask request

Protocol link: 10.11.1

The MRM-Permission-Ask request primitive is issued by the node controller of any node within a conference room that wishes to obtain permission for an action which impacts upon the room.

Figure 26 shows the sequence of events when using this primitive.

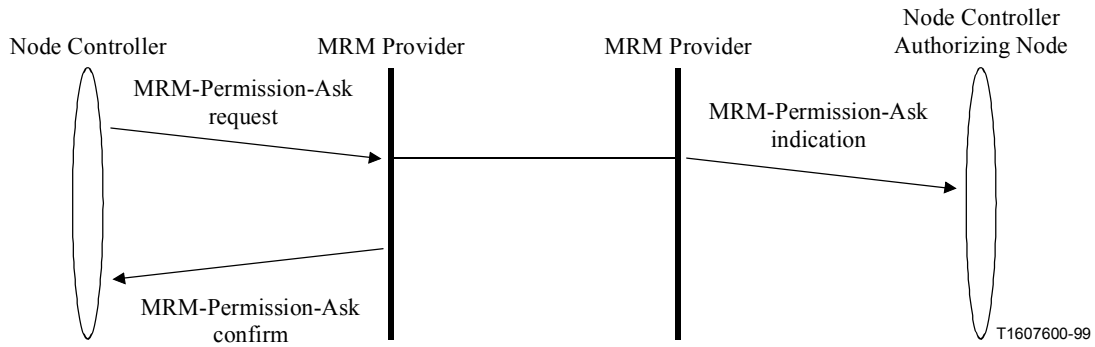


Figure 26/T.137 – MRM-Permission-Ask Request: User initiated – Sequence of primitives

The MRM Top Provider may initiate an MRM-Permission-Ask when it detects that a permission for a given action is required from an active role holder. (See Figure 27.)

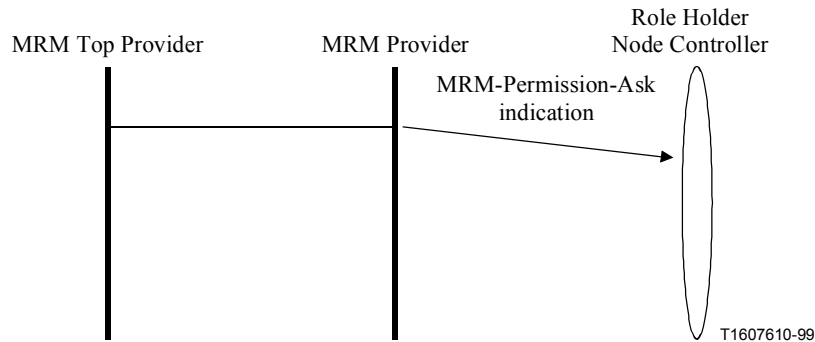


Figure 27/T.137 – MRM-Permission-Ask – Top Provider Initiated – Sequence of primitives

Table 24 shows the parameters and types of this primitive.

Table 24/T.137 – MRM-Permission-Ask Request – Types of primitives and their parameters

Parameter	Request	Indication	Confirm
Request Handle	M		M(=)
Requesting Node MRM User ID		M	
MRM Room Identifier	M	M(=)	
Permission Requested Handle	O	O(=)	
Group Permission Request	O	O(=)	
Associated Service	O	O(=)	
Result			M

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node MRM User ID: User ID of the requesting terminal.

MRM Room Identifier: The identifier for the MRM room.

Permission Requested Handle: Choice of: Enter room, Request to speak/Request to be seen/Request to use the whiteboard/Request to share an application.

Group Permission Request: Group ID – Used if the requester is making a request on behalf of the specified group to which it belongs.

Associated Service: The means to associate an MRM standardized service or a user-defined service.

Result: An indication of whether the request was accepted or rejected, and, if rejected, of the reason why. It contains one of the following possible results: Successful, Refused by chair.

9.9.1.2 MRM-Permission-Relinquish

Protocol link: 10.11.2

This primitive is issued by a node that has previously obtained permission for an action such as requesting the floor, either to cancel a pending request or to request to be removed from current floor activity.

The sequences of primitives belonging to both situations are depicted in Figure 28. Table 25 shows the parameters and types of this primitive.

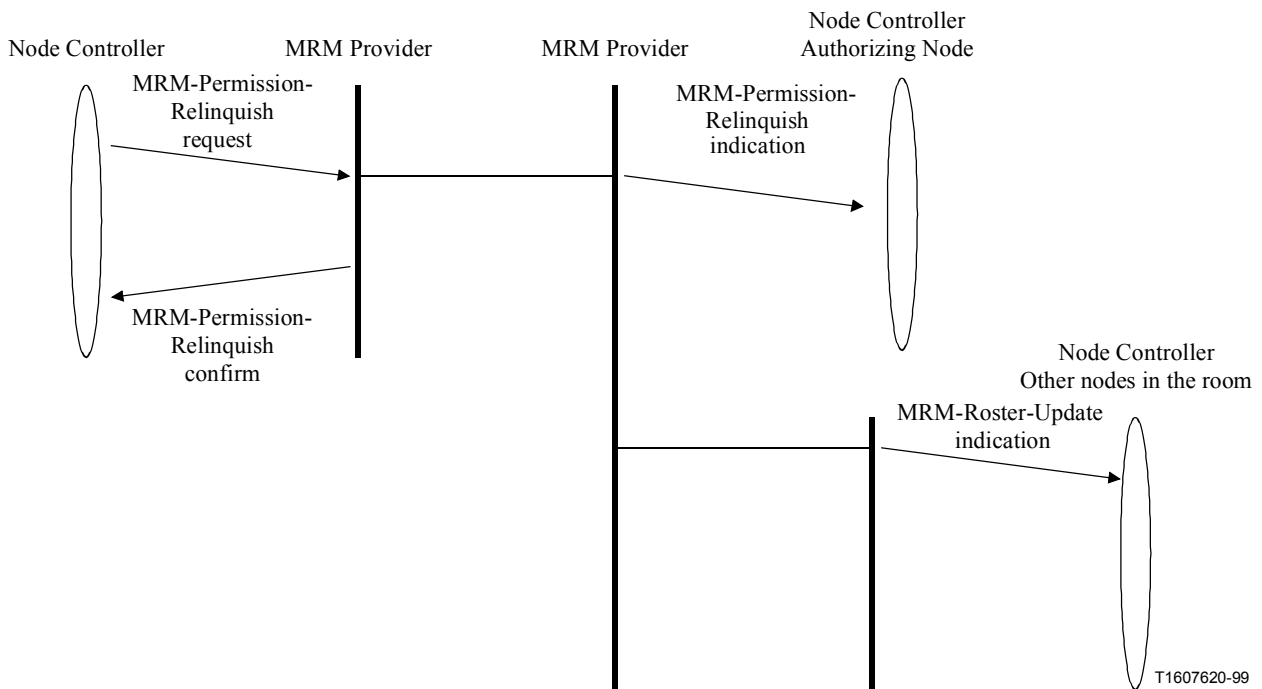


Figure 28/T.137 – MRM-Permission-Relinquish – Sequence of primitives

Table 25/T.137 – MRM-Permission-Relinquish – Types of primitives and their parameters

Parameter	Request	Indication	Confirm
Request Handle	M		M(=)
Requesting Node MRM User ID		M	
MRM Room Identifier	M	M(=)	
Release Reason	O	C	
Permission Relinquished	M	M	
Group Permission	O	O(=)	
Associated Service	O	O(=)	
Result			M

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node MRM User ID: Identifier of the node requesting the release of the floor.

Release Reason: This parameter can provide additional information within indication messages concerning the release, e.g. chair initiated, or system initiated.

Permission Relinquished: Permission identifier.

Group Permission Relinquished: Used if the requester is making a request on behalf of the specified group to which it belongs.

Associated Service: Audio Mixing is assumed by default; if a parameter is provided, it overrides the default.

Result: An indication of whether the request was accepted or rejected, and, if rejected, of the reason why. It contains one of the following possible results: Successful, Invalid room.

9.9.1.3 MRM-Permission-Grant

Protocol link: 10.11.3

The MRM-Permission-Grant primitive is issued by the node controller of a room chairing node in response to a previous Permission-Ask request.

Figure 29 shows the sequence of events when using this primitive. Table 26 shows the parameters and types of this primitive.

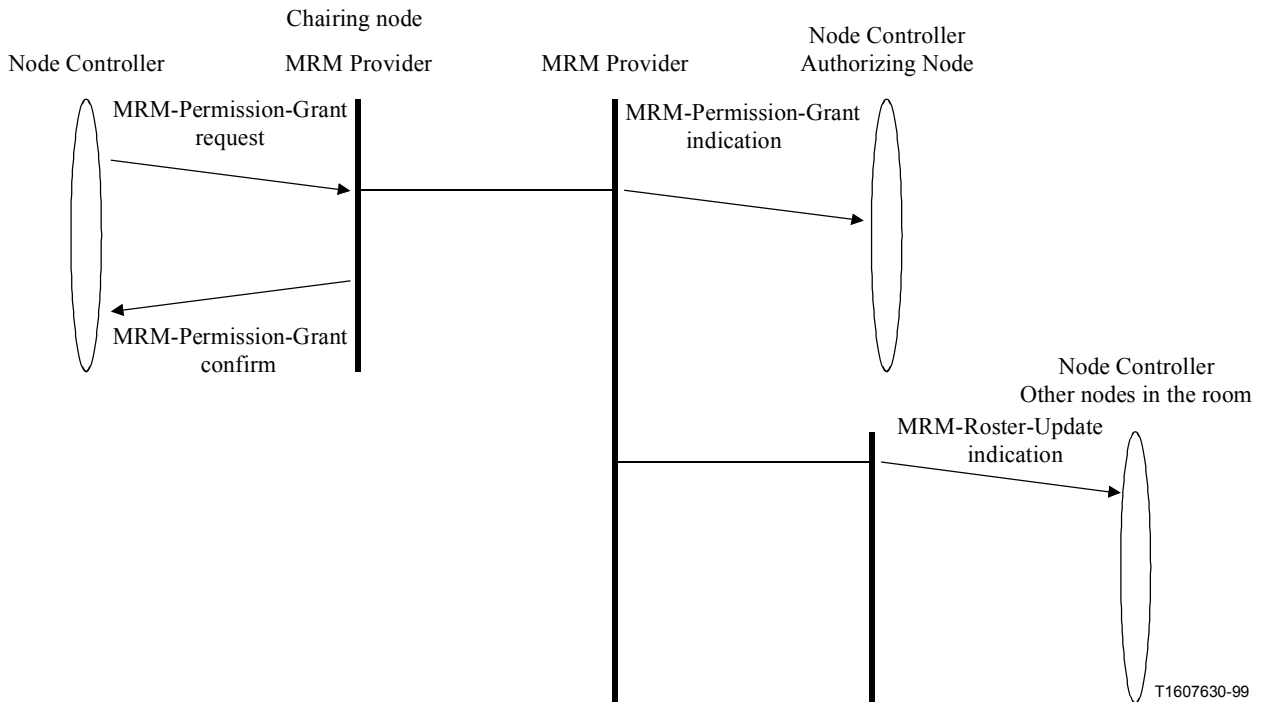


Figure 29/T.137 – MRM-Permission-Grant – Sequence of primitives

Table 26/T.137 – MRM-Permission-Grant – Types of primitives and their parameters

Parameter	Request	Indication	Confirm
Request Handle	M		M(=)
Requester's MRM User ID	M		
MRM Room Identifier	M	M=	
Permission Granted	M	M=	M(=)
Group Permission Grant	O	O=	
Associated Service	O	O=	
MRM Access Capability	O	O=	
Result			M(=)

Request Handle: Locally unique number to allow the requesting node to match responses.

Requester MRM User ID: Identification number of the node previously requesting the floor.

MRM Room Identifier: The identifier for the MRM room.

Permission Granted: Choice of permissions.

Group Permission Grant: Used if the requester is making a request on behalf of the specified group to which it belongs.

Associated Service: Audio Mixing assumed by default; if a parameter is provided, it overrides the default.

MRM Access Capability: List of functionality to which access has been granted.

Result: An indication of whether the request was accepted or rejected, and, if rejected, of the reason why. It contains one of the following possible results: Successful, Invalid request, Invalid room ID, Not authorized node.

9.10 Conference media controls and indications

MRM provides functions to assist in the management of real-time media used within MRM rooms. These include: the ability to identify an audio or a video source to the receivers, as well as the ability to tell a node sourcing audio or video that its stream(s) are being delivered to other terminals capable of receiving them; the ability to signal privacy on any of the media being used within a room. Service Commands allowing a node to start, subscribe or stop an MRM service.

The MRM protocol has no involvement in the establishment of real-time streams other than the ability to invite an audio participant to an MRM room. However, MRM provides mechanisms to support the unique identification of a media source within an MRM conference or room. It also provides a channel handle, a unique handle assigned by MRM using the MCS token mechanism. These handles can be mapped, by the MRM top provider and its network elements, onto the physical channels transporting the media, but this is beyond the scope of this Recommendation.

9.10.1 Description of abstract services

- MRM-Source-Identification – Message originating at a node sourcing audio and/or video to identify that source and the channel it is being transmitted on.
- MRM-On-Air-Indication – Indication to an audio or video source that its stream(s) are being delivered to one or more terminals capable of receiving them.
- MRM-Privacy-Notify – Allows a node to announce to the MRM Room that it is temporarily suspending participation to deal with a local matter.
- MRM-Service-Command – This primitive allows a node to Launch, Activate, Subscribe, or Deactivate an MRM service.

9.10.1.1 MRM-Source-Identification

Protocol link: 10.12

This function provides the capability for the origin and content of a real-time stream to be communicated to each recipient node.

Each node in the conference is uniquely identified by its MRM User ID (or represented by a proxy service). If a node is only capable of originating a single stream of a given media, then its User ID is all that will be needed to uniquely identify it as the originating node for that stream. Where a node is capable of originating more than one stream of a particular media, then this Recommendation supports the unique identification of streams. The MRM database provides the means to map streams to logical channels. Streams may be identified by tags embedded within the streams, or the identification may be implicit based upon a numeric ordering.

In a multipoint conference, when an MRM-aware node provides a media source to a real-time channel, it is required to send a source identification message to the conference management channel. If an identifier is not received from the node originating the stream, then the default audio

and/or video streams for that node will be assumed. In this case the MRM top provider will extract a name if available from the GCC conference roster entry for that node and generate the identification message itself. The identification message will allow a source to be associated with the real-time channel that is used to transport it. Where available the source identification information is used to replace the default determination of the source.

An MRM identification message can be associated with each received stream by the conference manager, or another network element acting on its behalf. It is the conference manager's responsibility to relay the identification onward to correspond to the routing applied to the stream. If any manipulation of a stream takes place, it must generate a new MRM Identification message to correlate with the changes.

The information will be used to identify to the receiver what is being sent and also from what location the stream originates. For a video stream it may be used to generate a caption for displaying with the image, but this is beyond the scope of this Recommendation.

Figure 30 shows the sequence of events when using this primitive. Table 27 shows the parameters and types of this primitives.

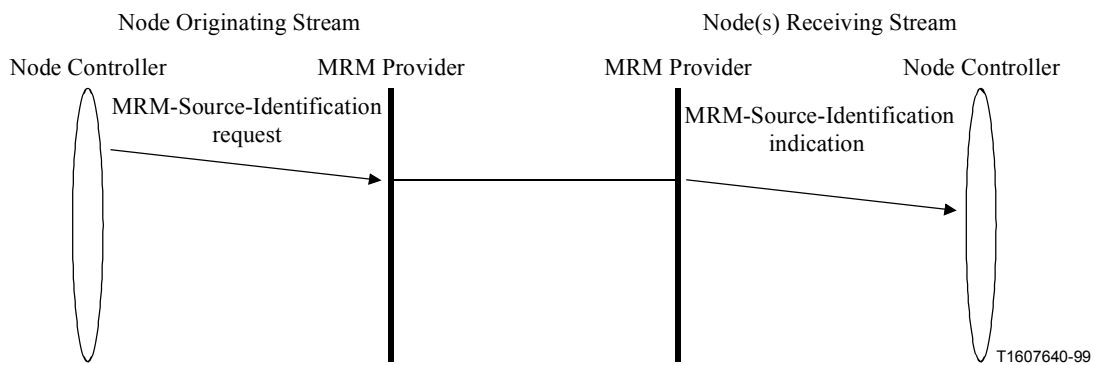


Figure 30/T.137 – MRM-Source-Identification – Sequence of primitives

Table 27/T.137 – MRM-Source-Identification Primitive

Parameter	Request	Indication
Source Handle	M	M(=)
Source Channel Identifier	M	M(=)
Media Type	M	M(=)
Stream Descriptor	O	O(=)
Associated Room ID	O	O(=)
Reason Code	O	O(=)

Source Handle: An MRM assigned handle which uniquely identifies the source by taking the node's MRM User ID and appending to it a locally unique ID to distinguish between different sources at the same node.

MEDIA Channel Identifier: The identifier for the channel the stream is being sent on. A dynamic MCS channel used as a unique identifier for the channel.

Media Type: Audio/Video/Other.

Stream Descriptor: Human- readable description of stream content.

Associated Room ID: Identifier for the MRM room to which stream is directed.

Reason Code: This code tells receiving site why image has changed – Mode change, Multipoint Channel Routing Override.

Terminal action

In order for a terminal to join the Identification service, it must send out its own video or audio identification message. Once this has been received, the node will be included in the service. Whenever the source it provides to the conference changes, it must send a new identification with the source device and identifier string. If the optional fields are not completed, the service will assume the source is the default video and/or audio streams and will use the user name string provided by the terminal's "GCC_Announce_Presence" message.

Network element actions

Each network element/MCU must be able to associate each stream it receives with a stream identifier. When a stream identifier is received for a stream arriving at one of its inputs the MRM provider must store the identifier and whenever a network element/MCU provides a new video source to any of its outgoing ports, it must send an MRM-Source-Identification primitive directed to the node receiving its output. Each network element needs to know the User ID of the terminals or network elements directly connected to its ports. If a connection is to another network element, it must also know the user ID of all other terminals and ports that can be reached via that MCU in order that it can route the messages using their User IDs. Each network element/MCU maintains its own local Stream Identification Register providing it with identifiers for each logical channel that it has as an input.

9.10.1.2 MRM-Privacy-Notify

Protocol link: 10.13.1

When a node wishes to temporarily suspend transmission of one of its real-time media streams, the MRM-Privacy-Notify message is sent from that endpoint to the conference management channel and the source removes or mutes its outgoing streams (unless it has asked for this to be done centrally). The conference manager forwards the privacy indication to those endpoints that were receiving the stream; however, it may also interpret the message itself and, for example, replace a video stream with a blue screen with a suitable message to the receiver(s). Likewise, when the privacy mode is cancelled by the endpoint, the procedure is reversed.

Figure 31 shows the sequence of events when using this primitive. Table 28 shows the parameters and types of this primitives.

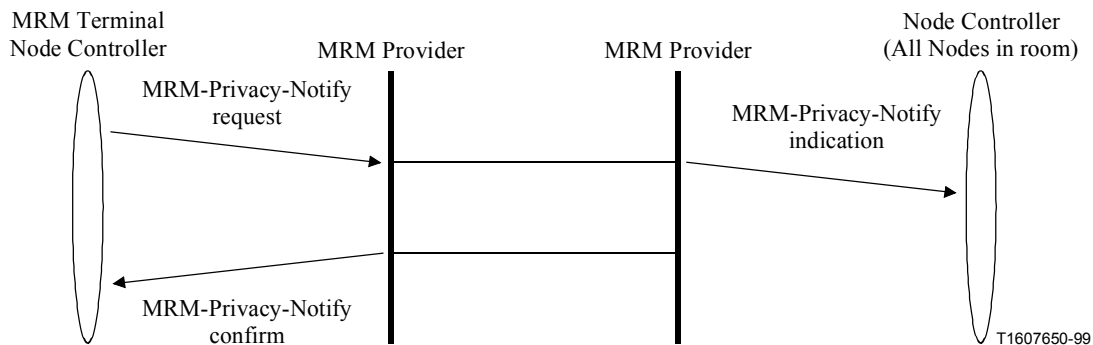


Figure 31/T.137 – MRM-Privacy-Notify – Sequence of primitives

Table 28/T.137 – MRM-Privacy-Notify Primitive

Parameter	Request	Indication	Confirm
Source Handle	M	M(=)	M(=)
Media Channel ID	M	M(=)	M(=)
Room ID	O	O(=)	O(=)
Invoke Privacy Remotely	M	M(=)	M(=)
Privacy Status	M	M(=)	M(=)

Source Handle: An MRM assigned handle which uniquely identifies the source by taking the node's MRM User ID and appending to it a locally unique ID to distinguish between different sources at the same node.

Media Channel ID: The identifier for the channel the stream is being sent on. A dynamic MCS channel used as a unique identifier for the channel.

Media Type: Audio/Video/Other.

Room ID: The ID for the room in which the privacy indication is relevant.

Invoke Privacy Remotely: Request for network to invoke privacy on behalf of requester.

Privacy Status: On/Off.

9.10.1.3 On-Air-Indication service

Protocol link: 10.13

The On-Air-Indication service is used in a multipoint conference to provide an indication back to a node originating a stream. Whenever a stream is routed to a terminal capable of interpreting it, then the source providing that stream is deemed to be On-Air. Only the conference manager or network elements acting in conjunction with the conference manager may generate the On-Air-Indication primitive.

The MRM Provider at each network element is responsible for maintaining a record of the On-Air status of each of its input channels. When the node controller at a network element makes a change to its outputs, or it receives an On-Air indication, and this results in a change of On-Air status of one of its inputs, then that network element should generate an MRM-On-Air-Indication request directed at node providing that stream to it.

When a source is routed by network elements to one or more receivers, then this event will result in the source node receiving an MRM-On-Air-Indication indication with the On-Air status flag set to TRUE. Only when the last receiver is removed will the sourcing node get another MRM-On-Air-Indication indication with the On-Air status flag set to FALSE.

Figures 32 and 33 show the sequence of events when using this primitive. Table 29 shows the parameters and types of this primitives.

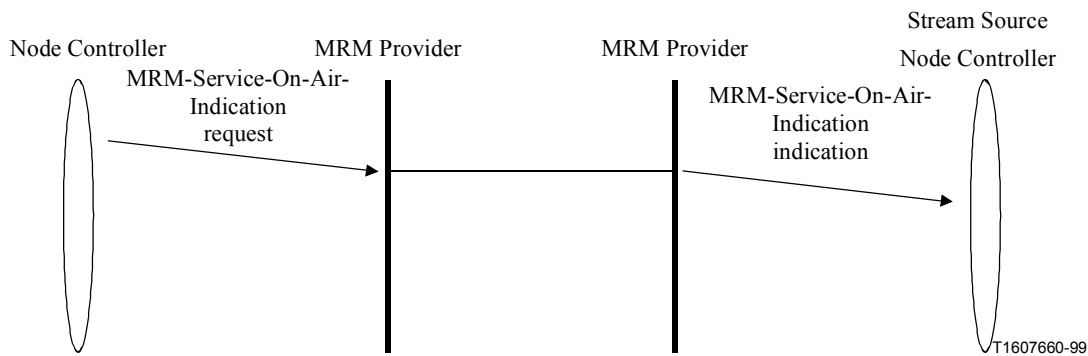


Figure 32/T.137 – MRM-On-Air-Indication – Sequence of primitives

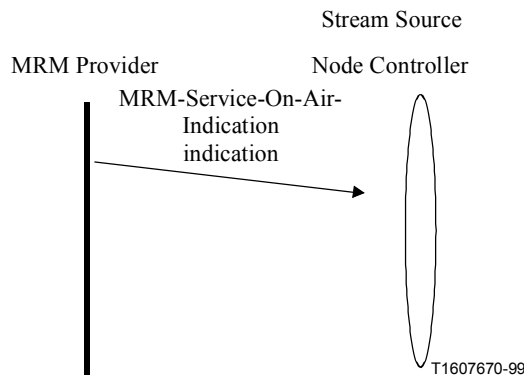


Figure 33/T.137 – Service-initiated MRM-On-Air-Indication – Sequence of primitives

Table 29/T.137 – MRM-On-Air-Indication primitive

Parameter	Request	Indication
Source Handle	M	M(=)
Media Channel Identifier	M	M(=)
Media Type	M	M(=)
Room ID	O	O(=)
On-Air Status Flag	M	M(=)

Source Handle: An MRM-assigned handle which uniquely identifies the source by taking the nodes MRM User ID and appending to it a locally unique ID to distinguish between different sources at the same node.

Media Channel Identifier: The identifier for the channel the stream is being sent on. A dynamic MCS channel used as a unique identifier for the channel.

Media Type: Audio/Video/Other.

Room ID: Identifier for MRM room to which stream is directed.

On Air Status Flag: TRUE/FALSE.

9.10.1.4 MRM-Service-Command

Protocol link: 10.14

The MRM-Service-Command request primitive enables a node controller at an MRM terminal or network element to send a service-related command to the conference manager, or for the conference manager to send a service-related command to other network elements participating in the delivery of the MRM conference. Participants and observers may, subject to their role privileges, subscribe to a service specified as offered in the MRM roster. Nodes shall subscribe to a service that is specified as required in the MRM roster. Nodes may also un-subscribe to a service that they voluntarily joined to, but no longer require.

The conference manager launches a service in response to framework requirement using the Launch form of the Service Command.

Figure 34 shows the sequence of events when using this primitive. Table 30 shows the parameters and types of this primitives.

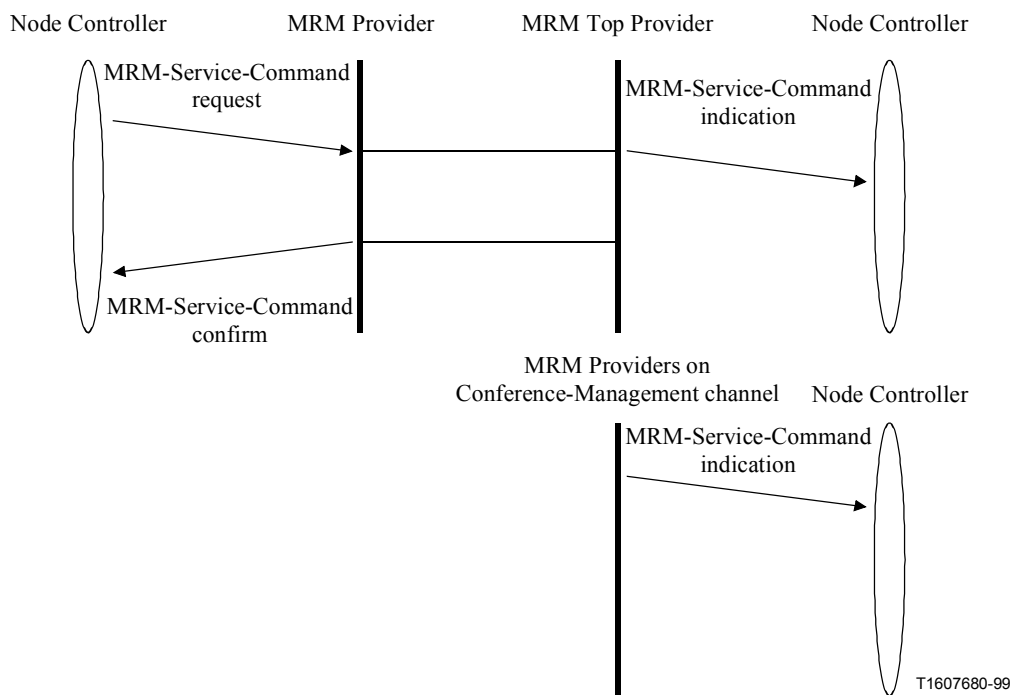


Figure 34/T.137 – MRM-Service-Command – Sequence of primitives

Table 30/T.137 – MRM-Service-Command primitive

Parameter	Request	Indication	Confirm
Request Handle	C	M	M(=)
Requesting Node		M	
MRM Room ID	C	M	
MRM Service Command	C	M	
Service Name ID	C	M	
Service Mode	O	O	
Participation Options	O	M	
Participant and Resources List	O	O	
Include Anonymous Nodes – Flag	O	O	
Result			M

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node: MRM User ID of the node issuing a service command.

MRM Room ID: Identifier for the MRM room in which the service command is to take effect.

Service Command ID: Choice of:

- *Launch* – Issued by conference manager to initiate network-based service functionality.
- *Subscribe* – Participants subscribe to an active service that they are not automatically subscribed to.
- *Unsubscribe* – Participants leave an active service that they no longer wish to be subscribed to.

Service Name ID: The MRM Service ID Key which the requesting node wishes to issue the command for.

Service Mode: This parameter is application specific.

Participation Options: This parameter allows the conference manager to specify a room-wide requirement.

Consists of the following choices:

- All participants at "service capable" nodes may join.
- All participants at "service capable" nodes shall join.
- Participation is specified node by node in the Participants and Resources List.

Participants and Resources List:

Choice of:

- *Node by node specification list* – List of participating terminal Node IDs with requirements for number and type of media channel, together with the logical channel specification.
- *Or a room requirement specification* which will be applicable to all nodes which participate:
 - *Source Nodes:* Optional list of nodes to receive from service.
 - *Number of source channels per node:* Optional number of sources per terminal node.
 - *Receiving Nodes:* Optional list of nodes to receive from service.
 - *Number of receive channels per node:* Optional number of receiving channels per terminal node.

Include Anonymous Nodes – Flag: Flag indicating anonymous nodes should be included as receivers.

Service Mode: The service mode is specified as a key; the default operation of the service will be assumed if not specified. The integer values in the range 1-255 are available for assignment within the individual services as required.

Import Source Identifier: This parameter allows a source from another room or a participating server to be included in the service. Each service that makes use of this facility describes how it is to be used.

Include Anonymous Nodes: Flag to Service if anonymous nodes can participate; this parameter is effectively service specific as it depends on the nature of the service as if it is appropriate to include anonymous nodes and if there are any participating.

Result: Indicates whether the request was accepted or rejected. It contains one of the following results: Successful, Service not available, Not authorized node, Invalid room ID.

9.11 MRM media services

MRM provides the means to define services supporting media use within rooms. It allows media to be integrated seamlessly into an MRM room. Currently, only Audio Bridging is standardized, providing a standardized solution to audiographic conferencing⁸. MRM also supports the definition of user-defined services.

9.11.1 Service events

The following service events require action from the MRM top provider or the network element acting on its behalf for a given service. Action for each of these events must be specified with the service definition.

Launching a media service

The MRM top provider is the only node within the MRM conference that can launch an MRM media service. The launch results from a framework requirement or as a result of a launch being requested by the convenor or chair for the room that the service is to be launched in.

Subscribing to a media service

If the MRM roster advertises an active MRM service where the subscription is not automatic for capable nodes, then a node has the ability to subscribe to the service using the subscribe service command.

Unsubscribing from a media service

A node that has chosen to subscribe to a service may also choose to unsubscribe from that service.

Entering an MRM room

When a node enters an MRM room, it becomes subject to the service regime that is in operation for that room. The participant options for the service specify what the requirement is for active services.

Leaving an MRM room

When a node leaves a room, any services that the node has subscribed to or has been subscribed to must be unsubscribed; this is the responsibility of the MRM top provider.

⁸ Other standardized services are expected in future revisions of this Recommendation.

9.11.2 Audio Bridging Service

The Audio Bridging service, like all MRM functionality, is room based and for each MRM room within a conference there is the potential for a separate Audio Bridge service to operate. The service specifies controls and indications between participants and the service. Inter-bridge control and signalling is currently beyond the scope of this Recommendation.

Default operation

When a terminal node enrolls its MRM application it specifies its audio capabilities. When a node enters an MRM room the MRM roster is sent to it; this roster indicates if the Audio Bridging Service is required. Any participant node that is audio capable will automatically be included in a required Audio Bridging Service.

The Audio Bridging service by default mixes the default stream from each audio-capable node in the room participating in the service. The service provides an audio output for each participating node which is a mix of the inputs; the algorithms for mixing are beyond the scope of this Recommendation. Service reporting is enabled and service participants in the room will be notified by service reports each time a supported event such as New Speaker detection is recorded by the service.

9.11.2.1 Chair-controlled rooms

In the absence of framework requirements, a chair role may be requested and assigned within a room as required. Once a chair has been assigned, it has the option to launch floor control as required.

Once floor control is launched, other MRM services are notified by the MRM top provider. On receiving an indication of Floor Control becoming active, the Audio Bridging Service changes its operational mode to Floor controlled. The framework determines the operation, but by default all audio inputs to all mixers are muted except those of the chair and of any participating operator. Participants may request to speak: the floor control service will direct these requests to the chair who can grant or refuse them or put them in a pending queue. Granted requests and relinquish floor requests from the chair are accompanied by a Set-Audio-Mix request being sent to the service; this request configures the Room mixers to what has been specified by the chair.

Where there is a defined framework, this may call for the service to be launched automatically as soon as the chair joins the room; in either case, the operation is the same.

9.12 Audio Mixer management

9.12.1 Description of abstract services

- MRM-Audio-Mixer-Set – This primitive is used to configure the audio mixer.
- MRM-AudioMix-Report – When enabled, this primitive provides status reports from the audio mixer.
- MRM-Invite-Audio-Only-Node – Allows an MRM node to issue an Invite that will cause an audio phone call to be made.
- MRM-Eject-Audio-Only-Node – Allows an MRM node to issue an Eject that will cause the audio line to be disconnected.
- MRM-Call-Status – Provides feedback to the initiator of an Invite requiring a dialled connection and is used to provide status information to the initiator of an MRM-Invite-Audio-Only-Node request.

9.12.1.1 MRM-Audio-Mixer-Set

Protocol link: 10.15

The MRM-Audio-Mixer-Set primitive provides the means for an authorized node to specify which audio streams are to be included in the audio mix for an MRM room. These requests are sent to the service management channel, where they are validated. If the requester is authorized to control the mixer the service manager constructs and issues appropriate out-of-band commands to control the mixer or mixers.

It may be the case that multiple network elements with separate bridges cooperate to deliver the mixing service. No explicit control and indications are specified in this MRM Recommendation version; however, where in-band signalling is needed, the Service Command primitives can be extended with user-defined parameters to address this need.

The requester may specify the required Bridge settings in a number of ways: A complete list of streams to be mixed can be specified, or only a list of streams to add to the current mix. A list of streams to be removed can be specified. Removing streams will free mixer resources. The mute and enable options should be used to address temporary changes, as is the case with floor control.

Figure 35 shows the sequence of events when using this primitive. Table 31 shows the parameters and types of this primitive.

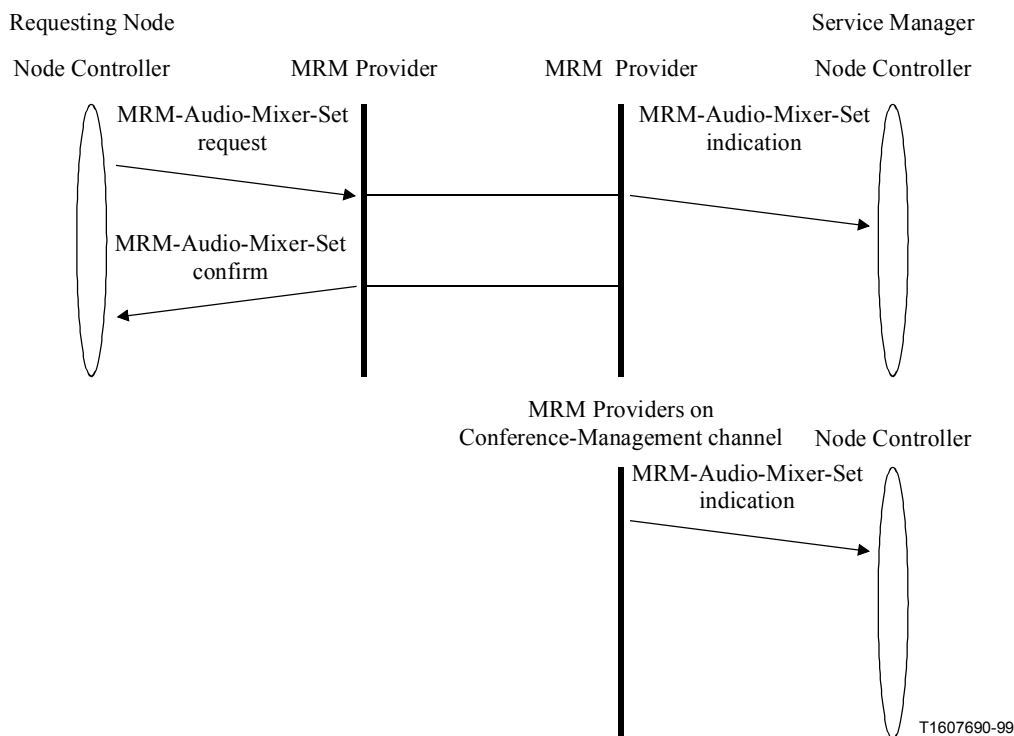


Figure 35/T.137 – MRM-Audio-Mixer-Set – Sequence of primitives

Table 31/T.137 – MRM-Audio-Mixer-Set – Types of primitives and their parameters

Parameter	Request	Indication	Confirm
Request Handle	M		M(=)
Originating Node MRM User ID		M(=)	
MRM Room Identifier	M	M(=)	
Audio Mixing Mode	O	O(=)	
Add Node-Stream (List)	O	O(=)	
Remove Node-Stream List	O	O(=)	
Enable Node-Stream (List)	O	O(=)	
Mute Node-Stream (List)	O	O(=)	
Include Node Stream (List)	O	O(=)	
Request Audio Level Adjust	O	O(=)	
Result			M

Originating Node MRM User ID: Requesting Node MRM User ID.

MRM Room Identifier: The identifier for the MRM room.

Mode: Manual/Default (distributed)/Centralized.

Add Node-Stream (List): List of streams to be added to an existing mix.

Remove Node-Stream (List): List of streams to be removed from an existing mix.

Enable Node-Stream (List): A list of streams to be enabled, i.e. to have a previous mute removed.

Mute Node-Stream (List): A list of streams to be muted at the mixer.

Include Node Stream (List): A complete list of all the streams to be included in the mix.

Remove Node-Stream (List): A complete list of streams to be removed from the mix.

Request Audio Level Adjust: Where this feature is available and permitted, it allows the requester to adjust the level of the mixer output delivered to it. By specifying another node, this command can be used to adjust the input to the requester's mix. The convenor or chair may adjust the levels for the entire Room.

Node MRM User ID + Device ID (if not default) – Optional; when not specified, mixer output will be adjusted.

Increase/Decrease 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%, 150%, 200%.

Flag Applies to all nodes in room – Only convenor or chair can set this flag.

Result: Request authorized, Request not authorized, No other network elements.

9.12.1.2 MRM-AudioMix-Report

Protocol link: 10.15.3

When the Audio Mixing service is configured with its reporting mechanism enabled, the MRM-AudioMix-Report primitive may be issued by the MRM service manager at the network element which is hosting the Audio Bridging service to nodes participating in the service.

A report is triggered by a service event such as "New speaker detected". Events in the Audio Bridging service may result from a participant speaking, an algorithm operating, or an MRM-Audio-Mixer-Set command; in all cases where announcements are enabled, a change in the audio mix

results in a new MRM-AudioMix-Report, which includes the node it has nominated to be the new speaker; it may also send its previous speaker assignment and a list of streams included in its mixer output.

Figure 36 shows the sequence of events when using this primitive. Table 32 shows the parameters and types of this primitives.

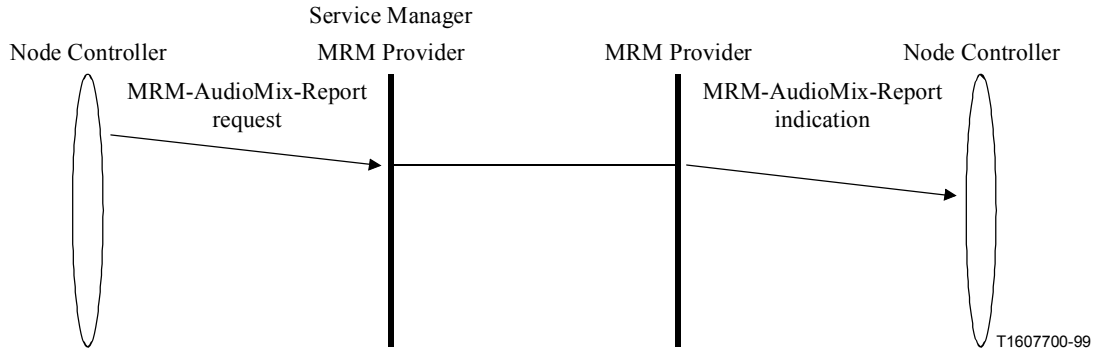


Figure 36/T.137 – MRM-AudioMix-Report – Sequence of primitives

Table 32/T.137 – MRM-AudioMix-Report – Types of primitives and their parameters

Parameter	Request	Indication
MRM User ID	M	M=
MRM Room Identifier	M	M=
New Speaker MRM User ID	M	M=
New Speaker Microphone Device ID	O	O=
Previous Speaker MRM User ID	M	M=
Previous Speaker Microphone Device ID	O	O(=)
List of Nodes included in mix	O	O(=)
List of Active Speakers	O	O(=)

MRM User ID: Detecting node MRM ID.

MRM Room Identifier: The identifier for the MRM room.

New Speaker MRM User ID: The MRM User ID of the node nominated locally as new speaker.

New Speaker Microphone Device ID: Where a terminal node has a number of participants with separate microphones and logical channels, these can be distinguished from the default audio stream for that node.

Previous Speaker MRM User ID: The MRM User ID of the node nominated locally as previous speaker.

Previous Speaker Microphone Device ID: Where a terminal node has a number of participants with separate microphones and logical channels these can be distinguished from the default audio stream for that node.

List of Nodes included in mix: List of Node User IDs, Mic IDs, Channel Ids identifying all streams being included in mix.

List of Active Speakers: List of all Node User IDs, Mic IDs, Channel IDs identifying as contributing to the output stream.

9.12.1.3 Audio Bridging service – Events

The MRM framework determines if the Audio Bridging service is available to the MRM conference. The MRM default framework does not include the service because it makes no assumptions about the requirement for audio. In services where audio is part of the baseline a new framework profile should be created and set to be the new default. When a framework is loaded that specifies a requirement for this service, there is a default behaviour defined. This specifies that the service will be launched and active and media capable nodes are required to subscribe to it. Loading a framework with a modified service requirement or participation requirement will modify this default behaviour. The MRM top provider or a network element acting on its behalf is responsible for launching, activating and de-activating services based upon framework requirements and also for dealing with service operation when a node enters or leaves an MRM room.

9.12.1.4 MRM-Invite-Audio-Only-Node

Protocol link: 10.15.4

The MRM-Invite-Audio-Only-Node primitive is used by suitably privileged participants to request that a participant be invited to join the audio part of the conference. This would be achieved by the MRM top provider issuing a dial-out request to the audio bridge (but this is beyond the scope of this Recommendation).

Figure 37 shows the sequence of events when using this primitive. Table 33 shows the parameters and types of this primitives.

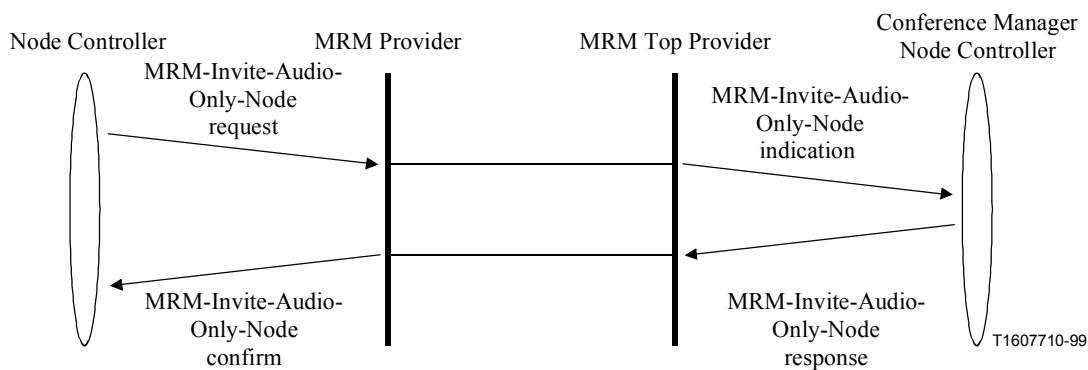


Figure 37/T.137 – MRM-Invite-Audio-Only-Node – Sequence of primitives

Table 33/T.137 – MRM-Invite-Audio-Only-Node

Parameter	Request	Indication	Response	Confirm
Request Handle	M			M(=)
DestinationRoom ID Handle	M	M(=)		
Node Communication Address	M	M		
NonT120node UserID			M	M(=)
Room Access Parameter	C	C(=)		
Reason for Invitation	O	O(=)		
Entry Method	O	O(=)		
Result			M	M

Request Handle: Locally unique number to allow the requesting node to match responses.

DestinationRoom ID handle: The Room ID for the virtual meeting space that the node is being invited to.

Node Communication Address: Choice of Phone Number, IP Address.

Room Access Parameter: For a Controlled access room, an entry ticket will be needed. This will be either:

 Password: TBD

Reason For Invitation: Optional string giving reason for invitation.

Entry Method: Choice of: Direct, By side conference.

Result: Indicates whether the request was successful. It contains one of the following results: Successful, Invalid Room ID, Not authorized node.

9.12.1.5 MRM-Eject-Audio-Only-Node

Protocol link: 10.15.4

The MRM-Eject-Audio-Only-Node primitive is used by suitably privileged participants to request that a participant be removed from the audio part of the conference. This would be achieved by the MRM top provider issuing a call terminate request to the audio bridge (but this is beyond the scope of this Recommendation).

Figure 38 shows the sequence of events when using this primitive. Table 34 shows the parameters and types of this primitives.

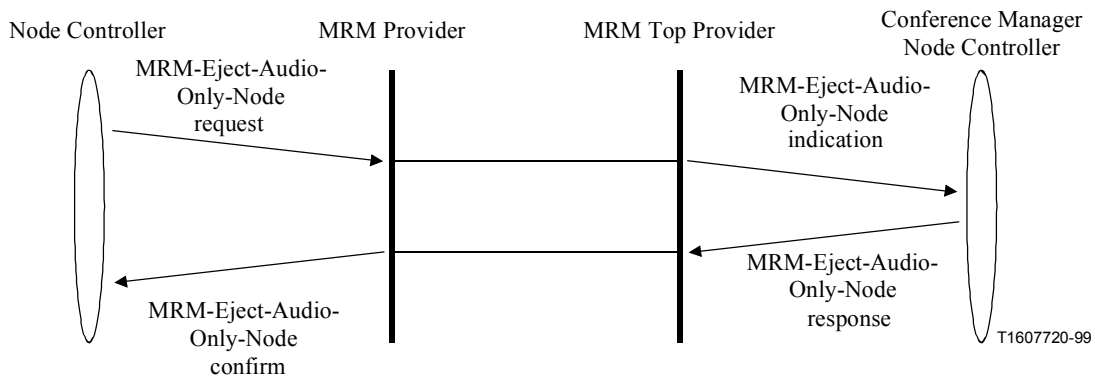


Figure 38/T.137 – MRM-Eject-Audio-Only-Node – Sequence of primitives

Table 34/T.137 – MRM-Eject-Audio-Only-Node

Parameters	Request	Indication	Response	Confirm
Request Handle	M			M(=)
Room ID Handle	M	M(=)		
NonT120node UserID	M	M(=)		
Result			M	M

Request Handle: Locally unique number to allow the requesting node to match responses.

Room ID Handle: The Room ID for the room that the node is participating in.

NonT120 UserID: User ID assigned to audio only node by the MRM top provider.

Result: Indicates whether the request was successful. It contains one of the following results: Successful, Invalid Room ID, Not authorized node.

9.12.1.6 MRM-Call-Status

Protocol link: 10.15.4

The MRM-Call-Status primitive is used by MRM top provider to signal back information on the progress of a dial-out connection.

Figure 39 shows the sequence of events when using this primitive. Table 35 shows the parameters and types of this primitives.

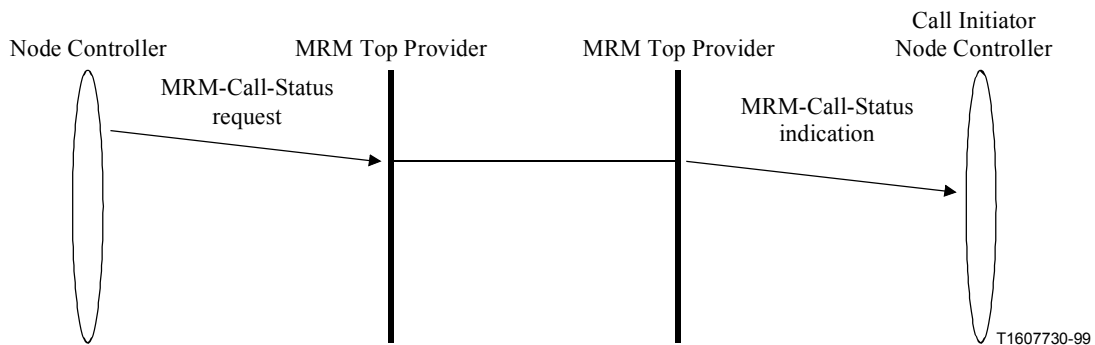


Figure 39/T.137 – MRM-Call-Status – Sequence of primitives

Table 35/T.137 – MRM-Call-Status

Parameter	Request	Indication
Room ID Handle	M	M(=)
NonT120node UserID	M	M(=)
Node Communication Address	M	M(=)
Status	M	M(=)

Room ID Handle: The Room ID for the room that the node is participating in.

NonT120 UserID: User ID assigned to audio only node by the MRM Top Provider.

Node Communication Address: Choice of Phone Number, IP Address.

Status: Dialling, Ringing, Engaged, NU, NoReply, Answered, Mixed.

10 MRM protocol definition

10.1 MRM node types

The MRM protocol supports two node types: Terminals and network elements. Terminals are effectively clients – users of the protocol. MRM also makes further distinctions between clients. A client may be either an observer or a participant. Network elements are management nodes that manage or contribute towards the management and/or delivery of the conference. For each of these categories the MRM initialization and operational procedure may differ. An MRM conference requires at least one node to participate in a management capacity. For this reason terminal-only nodes cannot launch MRM independently of network elements.

10.2 Launching MRM

The requirement for MRM at a Network Element Node Controller is signalled to its MRM Provider using the MRM-Service-Command primitive with the command set to "Launch". This primitive contains a flag that tell the MRM provider if it should compete for the MRM top provider role. If this flag is set, the MRM provider should store the Load-New-Framework Flag and follow the procedures defined in 10.3.

10.3 Determining the MRM top provider

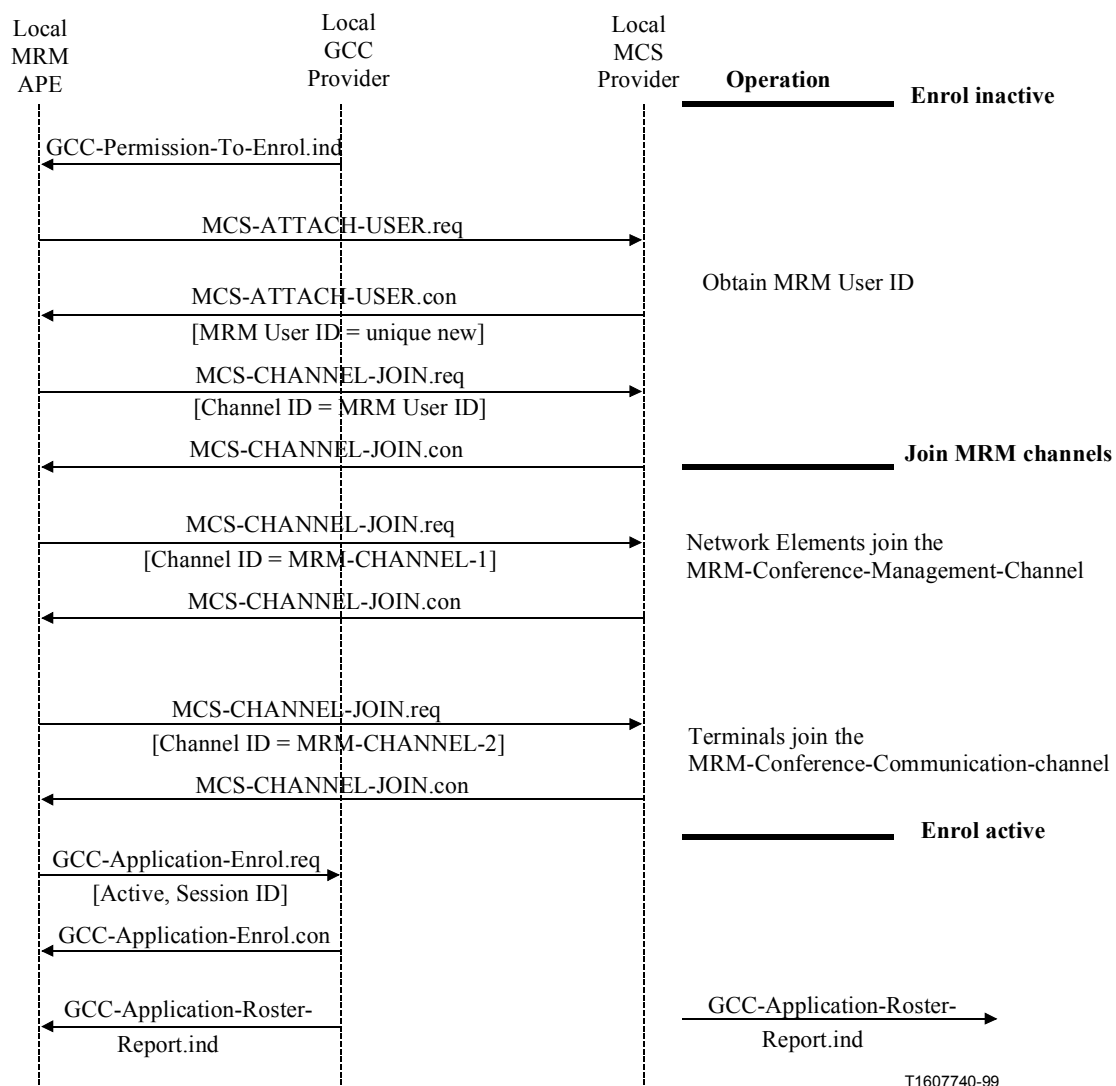
The node controller at the MRM top provider takes on the role of conference manager and is responsible for initializing and running the MRM conference. This top provider role is only open to nodes that participate in the conference as network elements. The principle use of MRM is expected to be in support of managed conferencing services, where the allocation of the MRM top provider role will often be known in advance, or it may even be hardwired into a system. In these cases the nominated node should grab the MRM-Top-Provider-Token; an MCS token assigned to represent the MRM top provider.

When the MRM top provider role has not been pre-assigned and the current GCC top provider has not indicated that it is MRM capable, then network elements may compete for it by attempting to grab the MRM-Top-Provider-Token. The node that succeeds in grabbing it is then nominated to become the MRM top provider.

The holder of the MRM-Top-Provider-Token shall join the MRM-Conference-Management-Channel.

10.4 MRM initialization and start-up

It is the task of the MRM top provider to initiate the MRM conference. Once a node has taken on the MRM top provider role it checks the "Load-New-Framework" flag provided in the earlier MRM-Launch request primitive. If it is set to TRUE, the MRM top provider must wait for an MRM-Framework-Load request to occur, or another MRM-Launch with the flag set to FALSE, before it can proceed. Once the new framework is loaded or the provider is given permission to proceed, it will enrol its MRM application with GCC, following the MRM Session Protocol Initialization sequence for a network element specified in Figure 40. This involvement is predefined as enrolling it as Inactive and obtaining an MCS User ID, the MRM ARM then joins the appropriate MRM channels for its MRM node type. The MRM top provider joins the MRM-Conference-Management-Channel. MRM requires only a single static session for operation of the MRM protocol. This session is initialized locally by the MRM top provider.



T1607740-99

Figure 40/T.137 – MRM session protocol initiation sequence

The MRM top provider has the task of executing the MRM framework requirement. It uses the MRM framework that it holds in its database, which will be the default or the default modified by a "convenor" loaded framework. First, it looks at the pre-defined room list and creates the specified rooms following the room creation procedures specified in 10.8.1. It looks at the service requirements for that room and launches and activates services as specified by the framework. Each service specifies its own operational procedure for events such as Launch and Subscription.

Once the MRM top provider is enrolled active within the conference, it appears in the GCC application roster. Once the other nodes see the MRM roster entry, they may then proceed to enrol by following the same Session Protocol Initialization sequence for their node type.

All network elements may join the MRM-Conference-Management-Channel⁹. This channel is used by terminals to communicate with the MRM top provider, and it also allows communication between the network elements when needed to support management of the conference and services. Terminals join the MRM-Conference-Client-Channel; once joined, they are effectively in the Lobby of the MRM Conference. The MRM top provider sends a RosterReportIndication directly to them on their MRM User ID channel at the highest data priority. This roster report contains conference- wide

⁹ This allows the possibility of other network elements shadowing the MRM Top Provider and provide a backup capability; however, this is beyond the scope of this Recommendation.

information pertaining to the MRM conference and includes a list of available rooms. Terminals may use the conference roster information to discover the room they wish to join or they may already have out-of-band information such as a room key that allows the MRM top provider to place them directly in the appropriate room. In either case a terminal wishing to Enter an MRM room issues an MRM-Room-Enter request, either providing a room key or the MRM Room ID; if it provides a Room ID it may also specify if it wishes to participate as an observer or as a participant. The resulting RoomEnterRequest PDU is sent to the MRM top provider on the MRM-Top-Provider-Channel and the Room Enter protocol procedures are followed. If the node is permitted to enter, it joins the appropriate room communications channel and sends a RoomReadyIndication to the MRM top provider. The RoomReadyIndication is the trigger for the MRM top provider to send the MRM room roster, which is sent to the node on its User ID channel at the highest data priority. Subsequent roster updates are sent as deltas of the original roster reports. These are sent on the appropriate broadcast channel for the conference or room, as specified in the roster update procedures.

See Tables 36 and 37.

Table 36/T.137 – Parameters for GCC-Application-Enrol request

Parameter	Contents
Conference ID	Provided by GCC-Application-Permission-To-Enrol indication
Session Key	{itu recommendation t 130 version(0) 1} and MRM-SESSION-ID (static)
Application User ID	Provided by MCS-Attach-User confirm
Active/Inactive	<i>All nodes initially enrol inactive until they obtain their MRM User ID (assigned by MCS)</i>
Conducting Operation Flag	This flag must not be set
Start-Up Channel	Static as specified in T.120: Network Elements join: MRM-Conference-Management-Channel Terminals join the MRM-Conference-Communication-Channel
Non-Collapsing Capabilities List	NodeType Flag: Terminal=TRUE; Network Element=FALSE ParticipationLevel Flag: Participant=TRUE; Observer=FALSE
Application Capability List	See Table below; omitted if Active/Inactive flag is set to Inactive
Enrol/Un-enrol	ENROL

**Table 37/T.137 – MRM ApplicationCapability List – Parameters for
GCC-Application-Enrol request**

Parameter	Parameter type	Contents For Terminal	Contents For Network Element
List of Media Services Supported	Logical	Set of Choice: PSTN-Audio H323-Audio H324-Audio H323-Video	Set of Choice: PSTN-Audio H323-Audio H324-Audio H323-Video Audio mixing Video Switching Video Streaming Shadow TopProvider

10.5 General Operation

All MRM PDUs are sent by the MRM provider issuing an MCS-Send-Data request with the MRM PDU in the data field. The MCS-Send-Data pPrimitive contains the initiator's User ID, the destination Channel ID and a Data Priority, which will be the highest data priority available unless otherwise specified. MRM is a management protocol like GCC, and has access to all the data priorities.

All MRM PDUs are categorized into either request, response, or indication classes. Request PDUs are defined to be those that require a corresponding response PDU in return. Indication PDUs are those that do not require a response (or in some cases, where the response is provided indirectly). For request PDUs which do not correspond to mandatory functionality, a generic response PDU (FunctionNotSupportedResponse) may be used and is specified in the GCC Recommendation (T.124), which allows the MRM provider receiving a request to respond to the requesting node without needing to have knowledge of the format of the specific response PDU. To allow the requester to know what this PDU is in response to, the entire request PDU is included within the FunctionNotSupportedResponse PDU. The FunctionNotSupportedResponse PDU shall be sent at the same priority level as that of the received request PDU.

NOTE – This terminology (request, response, indication) does not have a one-to-one correspondence to the definition of request, indication, response, and confirm used in the definition of the primitives. The terminology has been chosen to relate to the primary purpose of the PDU types with respect to the functions that they will perform. For this reason, the term "confirm" was not needed to describe PDUs, response PDUs are those which are sourced from a response primitive and result in a confirm primitive.

The MRM protocol includes support for non-standard extensions. On receipt of a nonStandardRequest PDU, an MRM Provider which does not understand the request shall issue a FunctionNotSupportedResponse PDU in return. A non-standard response or a non-standard Indication PDU may be ignored by an MRM provider.

10.6 Framework PDUs

FrameworkLoadIndication

Service link: 9.3

ASN.1 link: 11

A FrameworkLoadIndication is initiated by the MRM top provider when it detects that a node has joined the MRM conference as a network Element and specified the "Shadow Top Provider application capability". On receiving the GCC application roster report containing this capability, the MRM top provider will generate a FrameworkLoadIndication MRMPDU containing a copy of the active framework, specified as a list of parameters that modify the default settings, and send it on the MRM-Conference-Management-Channel at Top data priority. This capability is a logical collapsing capability and each time the node count increases for this capability the MRM top provider should re-send the PDU. Network elements not requiring the framework should ignore this PDU. (See Table 38.)

Table 38/T.137 – FrameworkLoadIndication MRMPDU

Content	Source	Sink
New Defaults Flag	MRM Top Provider	MRM Provider
Framework Originator Handle	MRM Top Provider	MRM Provider
Framework Handle	MRM Top Provider	MRM Provider
Conference Name	MRM Top Provider	MRM Provider
Conference Description	MRM Top Provider	MRM Provider
Conference Model	MRM Top Provider	MRM Provider
Enable Distributed Management	MRM Top Provider	MRM Provider
Primary Site ID	MRM Top Provider	MRM Provider
Slave Sites	MRM Top Provider	MRM Provider
Max Conventional Nodes Permitted	MRM Top Provider	MRM Provider
Max Anonymous Nodes Permitted	MRM Top Provider	MRM Provider
Permitted Services List	MRM Top Provider	MRM Provider
Permitted functions List	MRM Top Provider	MRM Provider
Permitted Roles List	MRM Top Provider	MRM Provider
Synchronize with Data Conf Flag	MRM Top Provider	MRM Provider
Groups Permitted Flag	MRM Top Provider	MRM Provider
Prohibited functions	MRM Top Provider	MRM Provider
Max number of Rooms	MRM Top Provider	MRM Provider
Max Depth of Rooms	MRM Top Provider	MRM Provider
Participant Conf Roster Reporting Flags	MRM Top Provider	MRM Provider
Participant Room Roster Reporting Flags	MRM Top Provider	MRM Provider
Observer Conf Roster Reporting Flags	MRM Top Provider	MRM Provider
Observer Room Roster Reporting Flags	MRM Top Provider	MRM Provider
Number of Pre-defined rooms	MRM Top Provider	MRM Provider
Predefined Room list	MRM Top Provider	MRM Provider
Room Constructor List	MRM Top Provider	MRM Provider

Table 38/T.137 – FrameworkLoadIndication MRMPDU (continued)

Content	Source	Sink
MRM Service Framework	MRM Top Provider	MRM Provider
Room 1		
Associated Framework Room ID	MRM Top Provider	MRM Provider
Framework Room ID	MRM Top Provider	MRM Provider
Room Name	MRM Top Provider	MRM Provider
Room Purpose Descriptor	MRM Top Provider	MRM Provider
Room Type	MRM Top Provider	MRM Provider
Room Model	MRM Top Provider	MRM Provider
Room Appearance	MRM Top Provider	MRM Provider
Room Function	MRM Top Provider	MRM Provider
Room Mode	MRM Top Provider	MRM Provider
Room Parent	MRM Top Provider	MRM Provider
Room Management Regime	MRM Top Provider	MRM Provider
Room Media	MRM Top Provider	MRM Provider
Room Access Regime	MRM Top Provider	MRM Provider
Child rooms Permitted	MRM Top Provider	MRM Provider
Max Participants Permitted	MRM Top Provider	MRM Provider
Max Observers Permitted	MRM Top Provider	MRM Provider
Secretary Required Flag	MRM Top Provider	MRM Provider
Secretary Access Key	MRM Top Provider	MRM Provider
Operator Required Flag	MRM Top Provider	MRM Provider
Operator Access Key	MRM Top Provider	MRM Provider
Groups Required List	MRM Top Provider	MRM Provider
Group Access Key(s)	MRM Top Provider	MRM Provider
Entry Regime	MRM Top Provider	MRM Provider
Exit Regime	MRM Top Provider	MRM Provider
Audio Support Required flag Audio only support	MRM Top Provider	MRM Provider
Floor control required flag Start-up configuration MRM Service list GCC App Jurisdiction List	MRM Top Provider	MRM Provider
Service Constructor Handle	MRM Top Provider	MRM Provider
Number of Audio channels per node	MRM Top Provider	MRM Provider
Number of Participants per node	MRM Top Provider	MRM Provider
Number of mixer inputs	MRM Top Provider	MRM Provider
Number of Audio Ports per Mix	MRM Top Provider	MRM Provider
Number of separate Audio Mixers	MRM Top Provider	MRM Provider
Required number of Inter NE Channels	MRM Top Provider	MRM Provider
Required Number of Terminal Access Channels	MRM Top Provider	MRM Provider

Table 38/T.137 – FrameworkLoadIndication MRMPDU (concluded)

Content	Source	Sink
Permissions to Control Mix	MRM Top Provider	MRM Provider
Start-Up Mode	MRM Top Provider	MRM Provider
Supported Mixing Modes Distributed – Management by speaker detection – Centrally – Manual	MRM Top Provider	MRM Provider
Supported features – Service Announcements – Floor Control – Stereo – Spatial – Surround	MRM Top Provider	MRM Provider

10.7 Roster PDUs

The MRM top provider is responsible for maintaining the MRM roster database; it is also responsible for issuing all RosterReportIndications, RosterUpdateIndications and responding to RosterEnquireRequests. All roster report and update PDUs contain a sequence number allocated by the MRM top provider. This mechanism allows the receiver to reconstruct the relative order of any roster information that it receives and thus avoid any potential of losing information or overwriting new information with old.

The MRM top provider maintains a separate number sequence for each channel it provides updates of. Each Update issued on a given channel causes that channel's sequence number to be incremented. Roster reports adopt the current sequence number for the channel that distributes its incremental updates. Roster reports do not cause the sequence number to increment and it is possible that a roster report and an update may contain the same sequence number; however, because updates increment the sequence, the recipient can be sure that the report is more recent than the update. The sequence number is a 16-bit number and if the upper limit is reached the sequence number resumes from zero.

MRM does not include nodes in terminal room rosters that are participating in a management role.

10.7.1 Roster Reporting

RosterReportIndication

Service link: 9.4.1.1

There are three cases that will trigger the MRM top provider to initiate a RosterReportIndication.

The first case is when a node joins the MRM conference and enrolls active in the MRM session. The MRM top provider as a member of the MRM session will receive the GCC-Application-Roster-Report for the MRM application. The top provider should update its MRM conference participants list from the GCC-provided information. If new nodes are detected, the MRM top provider sends each of them an MRM conference roster using a RosterReportIndication PDU, sent to their MRM UserID at Top data priority.

The second case is when a node enters an MRM room, the MRM top provider initiates a RosterReportIndication upon receiving an MRM-Room-Enter Response with the result "success" from the Conference Manager. The roster report contains a room roster and is sent directly to the node entering the MRM room on its User ID channel, sent at Top data priority.

In both of these cases, the content of the roster report is determined by the following criteria: the reason for the report, the default framework, or a new framework that has been loaded, and also the level of participation of the recipient node(s). MRM Participants and MRM observers have separate roster reporting specifications defined within the MRM framework that determine exactly what their roster reports will contain for a conference and for a room.

If the report results from a node joining the MRM conference, it will be sent conference roster information with the current sequence number from the MRM-Conference-Communication-Channel sequence. If the report results from a node entering an MRM room, it will be sent room roster information for the room being entered and the current sequence number drawn from the Room-Participant-Communication-Channel sequence. If a node is participating as an MRM observer, it may receive only a subset of the information provided to an MRM participant, delivered on the Room-Observer-Communication-Channel with the related sequence number.

The third case that will cause a RosterReportIndication is when the MRM top provider responds to a RosterEnquireIndication. In this case, the MRM top provider will construct the report based upon the information requested in the RosterEnquireIndication. It will then generate the RosterReportIndication and send it to the Enquirer's MRM User ID at High data priority.

In addition to the roster reports initiated by the MRM top provider the conference manager may initiate a roster report by sending an MRM-Roster-Report request specifying the required contents of the Report. The MRM top provider will construct the report based upon the information requested in the MRM-Roster-Report request. It will then generate the RosterReportIndication and send it to the specified channel at High data priority. (See Table 39.)

Table 39/T.137 – RosterReportIndication MRMPDU

Content	Source	Sink
Roster Sequence number	MRM Top Provider	Response
Scope of Report	MRM Top Provider	Response
Conference Configuration Information	MRM Top Provider	Response
Conference Status Information	MRM Top Provider	Response
Conference Participants List	MRM Top Provider	Response
Room Configuration	MRM Top Provider	Response
Room Status	MRM Top Provider	Response
Room ParticipantsList	MRM Top Provider	Response
Participant Details	MRM Top Provider	Response

On receipt of a RosterReportIndication PDU, the MRM provider shall first check the roster sequence number to ensure it is greater than that of any previously received roster report on its User ID channel or update sequence number for its associated update channel. If the sequence number is lower, then the MRM provider should check its store roster messages and re-order them into the right sequence. If the sequence number is higher but not consecutive with the previously received roster report or update, then it should be stored until the missing report or update arrives. If two roster PDUs arrive with the same sequence number, then one will be an update and the other will be a report. The update will proceed the report.

Once a correctly sequence roster report is received, the MRM provider generates an MRM-Roster-Report indication primitive and issues it to the Control MRMSAP. It may also update its local database if it maintains one.

10.7.2 Updating participants roster information

RosterUpdateIndication

The MRM top provider initiates a RosterUpdateIndication when information in its MRM roster database is updated and this results a change in parameters that have been previously included as part of a roster report or a roster update change. The MRM top provider constructs the RosterUpdateIndication PDU to contain only the changes from the last report or update and it inserts the roster sequence number into the PDU following the rules defined in 10.7. Conference-wide updates are sent on the MRM-Conference-Communication-Channel. If the report relates to a room, it will be sent either on the room's MRM-Room-Participant-Communication-Channel or on the MRM-Room-Observer-Communication-Channel. If the update is for network elements, it is sent on the MRM-Conference-Management-Channel. (See Table 40.)

Table 40/T.137 – RosterUpdateIndication MRMPDU

Content	Source	Sink
Roster Sequence Number	MRM Top Provider	Indication
Scope of Update	MRM Top Provider	Indication
Conference Configuration Information	MRM Top Provider	Indication
Conference Status Information	MRM Top Provider	Indication
Conference Participants List	MRM Top Provider	Indication
Room Configuration	MRM Top Provider	Indication
Room Status	MRM Top Provider	Indication
Room Participants List	MRM Top Provider	Indication
Participant Details	MRM Top Provider	Indication

When an MRM provider receives a RosterUpdateIndication, it shall first check the sequence number to ensure that it is higher than any previously received sequence numbers for the channel. If it is lower, then the provider should check stored updates and process them in the correct order, generating the MRM-Roster-Update indications in the correct sequence. If the update sequence number is higher but not consecutive with the previous sequence number, it should be stored until the missing update arrives. If the sequence number is the same as one previously received, then the update should be disregarded as it was issued prior to the roster report already received and processed.

Once a correctly sequence roster update is received, the MRM provider generates an MRM-Roster-Update Indication primitive and issues it to the Control MRMSAP. It may also update its local database if it maintains one.

10.7.2.1 Roster Enquiry

RosterEnquireIndication

Service link: 9.4.1.3

On receipt of an MRM-Roster-Enquire request primitive, the MRM provider shall send a RosterEnquireIndication PDU which contains the roster fields specified in the MRM-Roster-Enquire request. The MRM provider shall then send the RosterEnquireRequest PDU to the MRM-Conference-Management-Channel, specifying High data priority. (See Table 41.)

Table 41/T.137 – RosterEnquireRequest MRMPDU

Content	Source	Sink
Request Room List	Request	Top MRM Provider
Request Room templates	Request	Top MRM Provider
Room/channel ID	Request	Top MRM Provider
Request Room Configuration	Request	Top MRM Provider
Request Room Status	Request	Top MRM Provider
Request Room Participants List	Request	Top MRM Provider
Request Conference Configuration Information	Request	Top MRM Provider
Request Conference Status Information	Request	Top MRM Provider
Request Conference Participants Information	Request	Top MRM Provider

Replies to RosterEnquireRequests are made using the roster update mechanisms described above.

10.8 Room PDUs

10.8.1 Creating an MRM Room

RoomCreateRequest

Service link: 9.5.1.1

On receipt of an MRM-Room-Create request primitive, the MRM provider shall first check the Room Mode parameter. If the Mode is "dynamic" or "private", then the MRM provider shall obtain a dynamic MCS channel for the room by issuing an MCS ChannelConveneRequest. The ChannelConveneConfirm returns a channel ID is inserted into the Channel ID parameter, and a RoomCreateRequest PDU is constructed that contains the room creation parameters. It is sent to the MRM Conference Management-Channel at High data priority. If the Mode is "Static", the Room-Create will fail and the result will be set to "Invalid parameters" and an MRM-Room-Create confirm is immediately generated and issued to the Control MRMSAP. (See Table 42.)

Table 42/T.137 – RoomCreateRequest MRMPDU

Content	Source	Sink
Room ID Handle	Request	MRM Top Provider
Room Ownership Flag	Request	MRM Top Provider
Room Parent	Request	MRM Top Provider
Framework Room ID	Request	MRM Top Provider

Table 42/T.137 – RoomCreateRequest MRMPDU (concluded)

Content	Source	Sink
Use Template Flag	Request	MRM Top Provider
Room Template ID	Request	MRM Top Provider
Room Mode	Request	MRM Top Provider
Room Media	Request	MRM Top Provider
Room Access	Request	MRM Top Provider
Room Entry regime	Request	MRM Top Provider
Room Exit Regime	Request	MRM Top Provider
Room Name String	Request	MRM Top Provider
Room Purpose String	Request	MRM Top Provider
Number of Participants	Request	MRM Top Provider
Number of Observers	Request	MRM Top Provider
Required Roles List	Request	MRM Top Provider
Required Services List	Request	MRM Top Provider

The MRM top provider shall, on receipt of the RoomCreateRequest PDU, verify that the requesting node is authorized to create a room by checking the framework parameters MaxDepthOfRooms, MaxNumberOfRooms and prohibited PDUs. If any of the framework limits will be exceeded by accepting the request, or the requesting node is not authorized to send that PDU, then the request will be denied and the result will be returned informing the requester that the request has been refused and identifying the reason.

If the requesting node is authorized to create a room, the MRM top provider checks the RoomOwnerFlag. If the flag is set to FALSE, then the MRM Top provider issues an MCS ChannelConveneRequest to obtain a channel ID for the room; the MCS Channel ID becomes the MRM Room ID. If the RoomOwnerFlag is set to TRUE, then the request will provide an MCS channel for the room contained in the Room ID parameter, and the MRM top provider proceeds to create an MRM room entry in its database and associating it with the MCS Channel ID it has just obtained. The database is part of the MRM conference roster and the database entry may result in RosterUpdateResponses being sent if participating nodes have previously requested Conference Roster information or if the framework specifies it should be distributed. If the MCS-Channel-Convene fails, the MRM-Room-Create request will also be denied and a result will be returned identifying the failure reason that prohibits the request.

The MRM top provider also acts on framework requirements as part of its initialization procedure. This may involve creating and configuring MRM rooms according to the framework room constructor specification.

RoomCreateResponse

The top MRM provider responds to the RoomCreateRequest PDU by generating a RoomCreateResponse PDU which it sends to the requesting node. If the room was successfully created, the top MRM Provider will provide the MRM Room ID in the Room ID handle parameter field and create a master key for the room and then set the result to "Successful".

If the request is rejected, the MRM Room ID is set to "Null" and the result contains one of the following reason codes "out of MCS Resources", "Room Depth Exceeded", "Max Rooms Exceeded", "prohibitedByFramework". (See Table 43.)

Table 43/T.137 – RoomCreateResponse MRMPDU

Content	Source	Sink
Room ID handle	MRM Top Provider	Confirm
Room Master Key	MRM Top Provider	Confirm
Result	MRM Top Provider	Confirm

An MRM provider shall, on receipt of RoomCreateResponse subsequent to sending an RoomCreateRequest, generate an MRM-Room-Create confirm primitive containing the Room ID parameter and the result indicating if the request was successful or not.

10.8.2 Entering An MRM Room

RoomEnterRequest

Service link: 9.5.1.2

On receipt of an MRM-Room-Enter request primitive, the MRM provider shall construct a RoomEnterRequest PDU which contains the parameters supplied in the request plus the Room ID the participant node is currently in. If the node has just entered the conference, it will be placed in the lobby and the Room ID will be zero. The MRM provider shall then send a RoomEnterRequest PDU to the MRM-Conference-Management-Channel, specifying Top data priority. (See Table 44.)

Table 44/T.137 – RoomEnterRequest MRMPDU

Content	Source	Sink
Requesting node ID	MRM Provider	MRM Top Provider
Enter Room ID	Request/MRM Provider	MRM Top Provider
Current Room ID	MRM Provider	MRM Top Provider
Type of participation	Request/MRM Provider	MRM Top Provider
Terminate/Suspend Flag	Request/MRM Provider	MRM Top Provider
Room Access Key Parameter	Request/MRM Provider	MRM Top Provider
Reason for entry	Request/MRM Provider	MRM Top Provider

On receipt of a RoomEnterRequest, the MRM top provider must first do some basic checks on the room's ability to accommodate the entrant. It must check that the "RoomState" be either "Ready" or "Active", otherwise the result is set to "InvalidRoomID". Depending on the class of entrant, it must check the "MaxParticipants" or "MaxObservers" against the "NumParticipants" or "NumObservers" parameters to ensure that room limits will not be exceeded by the new entrant. If the limits are exceeded, the result is set to "RoomLimitsExceeded". Next, the MRM top provider shall validate the RoomAccessKey. If this corresponds to the master key for the room, then the top MRM provider will proceed to process the request by setting the result to "Successful", irrespective of the room access and entry regime.

If the master key was not provided, then the MRM top provider will check the entry regime for the room. If the entry regime is "Direct" and the RoomAccessModel is "Open" the MRM top provider can proceed to process the request by setting the result to "Successful". If the entry regime is "Direct" and the RoomAccessModel is "Controlled" the MRM top provider must check of the RoomAccessKey is valid for the room. If it is, then the result is set to "Successful"; otherwise, it is set to "NotAuthorized". If entry regime is "By Invitation", the MRM top provider will reject the request by setting the result to "PrivateRoom".

If the RoomEntryRegime is set to "ViaDoorKeeper", the MRM top provider will generate a PermissionAskPDU and send it to the node holding a "Door Keeper" role as specified in the room creation parameters or the framework room constructor. The PrivateRoomID in the PermissionAskRequest is set to a reception Room ID selected from a list of at least one reception room specified when the room was created. If no reception is free, the request will be held until a reception room and a "door keeper" are available. The "door keeper" should enter the reception room (setting the suspend flag so it will continue to receive state information from the main room). If the RoomEntryRegime is "ViaAutomatedReception", then the MRM top provider takes the Reception Room ID and puts it in the "Goto Room ID" parameter of the RoomEnterResponse PDU; it then sets the result to "GotoSpecifiedRoom".

The MRM top provider having processed the request now generates a RoomEnterResponse PDU containing the result and any Room IDs placed into the "Goto Room ID" parameter and sends it on the User ID channel of the requester at Top data priority.

If the result is successful, the MRM top provider will update the room roster, adding the requesting node to the room and removing it from the room it was previously in; at the same time it must check roles, groups and services for any required actions in leaving the previous room and joining the new room and follow the procedure specified there for leaving and entering MRM rooms.

RoomEnterResponse

Finally, the top MRM provider shall construct a RoomEnterResponse PDU containing the result of the RoomEnterRequest and send it to the MRM User ID of the node entering the room at Top data priority. (See Table 45.)

Table 45/T.137 – RoomEnterResponse MRMPDU

Content	Source	Sink
Enter Room Channel ID	MRM Top Provider	Confirm
Room Access Key	MRM Top Provider	Confirm
Channel to Join	MRM Top Provider	Confirm
Reason for entry	MRM Top Provider	Confirm
Result	MRM Top Provider	Confirm

The MRM provider that initiated the RoomEnterRequest shall on receipt of the RoomEnterResponse PDU generate an MRM-Room-Enter confirm primitive containing the result indicating if the request was successful or not. If the result is "Successful", then the MRM provider shall use the Channel ID returned in the Destination Room Channel ID parameter as the channel to be joined (if provided), and it will issue an MCS ChannelJoin request PDU in order to join the MCS channel for the room. The joining of the MCS channel is confirmed by MCS with the ChannelJoin confirm PDU and, providing this is successful, the MRM provider sends a RoomReadyIndication back to the MRM top provider at Top data priority.

RoomReadyIndication

The RoomReadyIndication PDU is only used as part of the Room Entry procedure to tell the MRM top provider that the node entering a room has joined the appropriate channels for the room and is ready to receive room roster information. (See Table 46.)

Table 46/T.137 – RoomReadyIndication MRMPDU

Content	Source	Sink
Room ID	MRM Provider	MRM Top Provider

Once the MRM top provider receives the RoomReadyIndication, it generates a RosterReport that contains a room roster for the room that the node has just entered. The RosterReport is sent directly to the node entering the room on its User ID channel at Top data priority. Once a node is in the room, it receives roster updates as deltas to the roster report on the room communication channel that it joined.

10.8.3 Invitation to an MRM Room

RoomInviteIndication

Service link: 9.5.1.3

On receipt of an MRM-Room-Invite request primitive, the MRM provider shall construct a RoomInviteIndication PDU which contains the parameters supplied in the request. The MRM provider shall then send a RoomInviteIndication PDU to each of the MRM User IDs specified in the Invited node list, specifying High data priority. The MRM provider then generates an MRM-Room-Invite confirm and issues it to its Control MRMSAP. (See Table 47.)

Table 47/T.137 – RoomInviteIndication MRMPDU

Content	Source	Sink
Room ID Handle	Request	Indication
Room Entry Key	Request	Indication
Invited Node List of MRM User IDs	Request	Indication
Invitor's MRM User ID	Request	Indication
Reason String	Request	Indication

On receipt of a RoomInviteIndication PDU, an MRM provider will generate an MRM-Room-Invite indication and send it to its node controller on the Control MRMSAP. Nodes wishing to accept the invitation use the MRM-Room-Enter procedure.

10.8.4 Creating a group

GroupCreateRequest

Service link: 9.6.3.1

On receipt of an MRM-Group-Create request primitive, the MRM provider shall construct a GroupCreateRequest PDU which contains the parameters supplied in the request. The MRM provider shall then send it to the MRM Top Provider on the MRM-Conference-Management-Channel specifying High data priority. (See Table 48.)

Table 48/T.137 – GroupCreatRequest MRMPDU

Content	Source	Sink
Group Owner's MRM User ID	Request	MRM Top Provider
Group Scope Flag	Request	MRM Top Provider
Room ID Handle	Request	MRM Top Provider
Group Name	Request	MRM Top Provider
Group Function	Request	MRM Top Provider
Ordered Group Flag	Request	MRM Top Provider
Members List	Request	MRM Top Provider

On receipt of a GroupCreateRequest PDU, the MRM top provider checks the creator has "group creation" role privileges. If it does, then the MRM top provider sets the result to "Successful". If the creator does not have the appropriate privileges then the result is set to "Not Authorized".

GroupCreateResponse

If the request is permitted, the MRM top provider obtains an MCS channel for the group and then makes the database entries for the group. In all cases the MRM top provider generates a GroupCreateResponse PDU containing the result, and sends it to the requesting node on its User ID channel at High data priority. A successful group creation results in an MRM roster update announcing the group. This also serves to inform the group members of their participation. (See Table 49.)

Table 49/T.137 – GroupCreateResponse MRMPDU

Content	Source	Sink
Room ID Handle	MRM Top Provider	Confirm
Group ID	MRM Top Provider	Confirm
Result	MRM Top Provider	Confirm

On receipt of the RoomDestroyResponse PDU the MRM Provider sends an MRM-Room-Destroy confirm containing the room reference and the result code.

GroupDisbandRequest

Service link: 9.6.3.2

On receipt of an MRM-Group-Disband request primitive, the MRM top provider shall construct a GroupDisbandRequest PDU which contains the parameters supplied in the request. The MRM provider shall then send it to the MRM top provider on the MRM-ConferenceManagement-Channel specifying High data priority. (See Table 50.)

Table 50/T.137 – GroupDisbandRequest MRMPDU

Content	Source	Sink
Group Owner	Request	MRM Top Provider
Room ID	Request	MRM Top Provider
Group ID	Request	MRM Top Provider

On receipt of a GroupDisbandRequest PDU, the MRM top provider checks that the requester either has "group create/modify/destroy" role privileges, or is the owner of the group. If it does, the group is deleted from the MRM conference database, the result is set to "successful" and an MRM-Group-Disband indication is sent on the Control MRMSAP to its node controller. The destruction of a group will trigger a RosterUpdateIndication to any node monitoring that information.

GroupDisbandResponse

The MRM Top Provider then generates a GroupDisbandResponse PDU containing the reference to the room requested to be destroyed and the result indicating if the request was successful or providing the reason why it failed. (See Table 51.)

Table 51/T.137 – GroupDisbandResponse MRMPDU

Content	Source	Sink
Group ID	MRM Top Provider	Confirm
Result	MRM Top Provider	Confirm

On receipt of the GroupDisbandResponse PDU, the MRM provider sends an MRM-Group-Disband confirm containing the room reference and the result code.

GroupMemberSpecifyRequest

Service link: 9.6.3.3

On receipt of an MRM-Group-Member-Specify request primitive, the MRM provider shall construct a GroupMemberSpecifyRequest PDU which contains the parameters supplied in the request. The MRM provider shall then send it to the MRM-Conference-Management-Channel specifying High data priority. (See Table 52.)

Table 52/T.137 – GroupMemberSpecifyRequest MRMPDU

Content	Source	Sink
Group Identifier	Request	Indication
Add Participants List	Request	Indication
Remove Participants List	Request	Indication
Specify Participants List	Request	Indication

On receipt of a GroupMemberSpecifyRequest PDU, the MRM top provider shall check that the requester either has "group create/modify/destroy" role privileges, or is the owner of the group. If it is authorized, the conference database is amended to reflect the request, and the result is set to "successful". The changes to a group's membership will trigger a RosterUpdateIndication to any node monitoring that information.

GroupMemberSpecifyResponse

The MRM TopProvider generates a GroupMemberSpecifyResponse PDU and sends it to the requester's User ID at High data priority. (See Table 53.)

Table 53/T.137 – GroupMemberSpecifyResponse MRMPDU

Content	Source	Sink
Group Identifier	Response	Confirm
Result	Response	Confirm

On receipt of a GroupMemberSpecifyResponse PDU, an MRM provider generates an MRM-Group-Member confirm and sends it to the Control MRMSAP to its node controller.

10.9 T.120 Session Association

SessionAssociateRequest

Service link: 9.7.1.1

On receipt of an MRM-Session-Associate request primitive, the MRM top provider shall construct a SessionAssociate PDU which contains the parameters supplied in the request. The MRM provider shall then send it to the MRM top provider on the MRM-ConferenceManagement-Channel specifying High data priority. (See Table 54.)

Table 54/T.137 – SessionAssociateRequest MRMPDU

Content	Source	Sink
Room ID	Request	MRM Top Provider
Associated Sessions	Request	MRM Top Provider

On receipt of a SessionAssociateRequest PDU, the MRM top provider first checks that the requesting node has privileges to modify room parameters. If it does, it makes an entry in the room roster for the specified room and sets the result to "Successful"; otherwise, it sets the result to "Not Authorized". A successful session association causes a RosterUpdate PDU to be sent to any node in the room monitoring this information.

SessionAssociateResponse

The MRM top provider then generates a SessionAssociateResponse PDU containing the result parameter and sends it to the requester on its User ID channel at Top data priority. (See Table 55.)

Table 55/T.137 – SessionAssociateResponse MRMPDU

Content	Source	Sink
Result	MRM Top Provider	Response

10.10 Room role PDUs

10.10.1 Assigning an MRM role

The MRM room roster will contain a list of roles available to the room that is passed to participant nodes when they enter the room. Each provider is required to maintain a list of active roles and role holders for the room that it is in. The chair role is available for use in the default framework specification.

RoomRoleAssignRequest

Service link: 9.8.2.1

On receipt of an MRM-Room-Role-Assign primitive, the MRM provider constructs a RoomRoleAssignRequest PDU from the request and inserts its nodes MRM User ID and sends it to the MRM top provider on the MRM-Conference-Management-Channel at Top data priority. (See Table 56.)

Table 56/T.137 – RoomRoleAssignRequest MRMPDU

Content	Source	Sink
Requesting Node's MRM User ID	MRM Provider	MRM Top Provider
Proposed Role Holders MRM User ID	Request	MRM Top Provider
MRM Room Identifier	Request	MRM Top Provider
MRM Room Role Identifier	Request	MRM Top Provider
MRM Role Access Key	Request	MRM Top Provider
Group Identifier	Request	MRM Top Provider

On receipt of a RoomRoleAssign PDU, the MRM top provider checks if the role requested is available for assignment by searching the MRM conference roster PermittedRoles list. If it is not permitted, the result is set to "RoleNotPermitted". If the role is permitted, the top MRM provider next checks if the role is already assigned. If it is, the result is set to "RoleAlreadyAssigned". If the role is permitted and available for assignment, then the MRM top provider checks the roster database, looking for any access key requirement associated with that role. If a key is required, the MRM top provider checks the MRM Role Access Key parameter in the request. If it holds a key, it attempts to validate the key against its database key. If the key is not validated, the result is set to "InvalidKey". If the requesting node meets all the conditions, then the MRM top provider updates the room roster, assigning the requesting node the role.

RoomRoleAssignResponse

On completion of processing the RoomRoleAssignRequest PDU, the MRM top provider constructs a RoomRoleAssignResponse PDU, inserts the result, and sends it to the MRM User ID of the requesting node at Top data priority. (See Table 57.)

Table 57/T.137 – RoomRoleAssignResponse MRMPDU

Content	Source	Sink
MRM Room ID	MRM Top Provider	Confirm
MRM Role	MRM Top Provider	Confirm
Result	MRM Top Provider	Confirm

On receipt of a RoomRoleAssignResponse, the MRM provider which initiated the request sends an MRM-Room-Role-Assign confirm containing the result indicating whether the request was successful or not.

10.10.2 Relinquishing an MRM role

RoomRoleReleaseRequest

Service link: 9.8.2.2

On receipt of an MRM-Role-Release request primitive, the MRM provider constructs a RoomRoleReleaseRequest PDU and sends it to the MRM top provider on the MRM-Conference-Management-Channel at Top data Priority. (See Table 58.)

Table 58/T.137 – RoomRoleReleaseRequest MRMPDU

Content	Source	Sink
MRM Room Identifier	Request	MRM Top Provider
Role Holder Identifier	Request	MRM Top Provider
MRM Room Role	Request	MRM Top Provider
Group ID	Request	MRM Top Provider
Release Reason	Request	MRM Top Provider

On receipt of a RoomRoleReleaseRequest PDU, the MRM top provider updates its room roster database removing the role assignment and then it generates a response to indicate that the request has been dealt with. The role release will trigger a RosterUpdateIndication to the nodes in that room that are monitoring this information.

RoomRoleReleaseResponse

The MRM top provider constructs a RoomRoleReleaseResponse PDU, inserts the result and sends it to the MRM User ID of the requesting node at Top data priority. (See Table 59.)

Table 59/T.137 – RoomRoleReleaseResponse MRMPDU

Content	Source	Sink
MRM Room Identifier	MRM Top Provider (from Request)	Confirm
MRM Room Role	Request (from Request)	Confirm
Result	MRM Top Provider	Confirm

On receipt of the RoomRoleReleaseResponse, the MRM provider that initiated the RoomRoleReleaseRequest generates an MRM-Room-Role-Release response primitive and sends it to the Control MRMSAP.

RoomRoleReleaseIndication

The MRM top provider constructs a RoomRoleReleaseResponse PDU, inserts the result and sends it to the MRM User ID of the requesting node at Top data priority. (See Table 60.)

Table 60/T.137 – RoomRoleReleaseIndication MRMPDU

Content	Source	Sink
Requesting Node MRM User ID	MRM Provider	MRM Top Provider
MRM Room Identifier	Request	MRM Top Provider
MRM Room Role Identifier	Request	MRM Top Provider
Group Identifier	Request	MRM Top Provider
Release Reason	Request	MRM Top Provider

On receipt of a RoomRoleReleaseIndication PDU, the MRM provider generates an MRM-Room-Role-Release response primitive and sends it to the Control MRMSAP.

10.10.3 Asking for an MRM role

RoomRolePleaseRequest

Service link: 9.8.2.3

On receipt of an MRM-Role-Please request primitive, the MRM provider constructs a RoomRolePleaseRequest PDU and, using the information on role holder User IDs in its local database, sends the PDU to the identified role holder's MRM User ID at High data priority. The MRM provider then generates an MRM-Room-Role-Please confirm primitive and sends it to its Control MRMSAP. (See Table 61.)

Table 61/T.137 – RoomRolePleaseRequest MRMPDU

Content	Source	Sink
Requesting Node MRM User ID	MRM Provider	MRM Top Provider
MRM Room Identifier	Request	MRM Top Provider
MRM Room Role Identifier	Request	MRM Top Provider
MRM Group Identifier	Request	MRM Top Provider

On receipt of a RoomRoleRequest PDU, the MRM provider checks whether it is the current holder of the role being requested. If it is not, it ignores the request. If it is the current holder of the requested role, it generates an MRM-Room-Role-Please indication primitive and sends it to its Control MRMSAP.

10.10.4 Passing An MRM Role to another Room Participant

RoomRoleGiveRequest

Service link: 9.8.2.4

On receipt of an MRM-Role-Give request primitive, the MRM Provider checks whether it is the current holder of the role to be given. If it is not, the result is set to "NotRoleHolder" and a confirm is immediately generated and sent to the Control MRMSAP with the result indicating the Give request failed. Otherwise, the MRM provider constructs a RoomRoleGiveRequest PDU and sends it on the MRM-Conference-Management-Channel at High data Priority. (See Table 62.)

Table 62/T.137 – RoomRoleGiveRequest MRMPDU

Content	Source	Sink
Role Holder MRM User ID	Request	Indication
Role Recipient's Node (MRM User ID)	Request	MRM Provider
MRM Room Identifier	Request	Indication
MRM Room Role Identifier	Request	Indication
Group Identifier	Request	Indication

On receipt of a RoomRoleGiveRequest PDU, the MRM provider checks whether the requester is the current role holder. If it is not, it sets the result to "NotRoleHolder". Otherwise, it generates an MRM-Room-Role-Give indication and sends it to its Control MRMSAP.

RoomRoleGiveResponse

On receipt of an MRM-Room-Role-Give response primitive, the MRM provider generates a RoomRoleGiveResponse PDU and sends it to the requester's MRM User ID at High data priority. (See Table 63.)

Table 63/T.137 – RoomRoleGiveResponse MRMPDU

Content	Source	Sink
MRM Room Identifier	Request	Indication
MRM Room Role Identifier	Request	Indication
Group Identifier	Request	Indication
Result	Request	Indication

On receipt of a RoomRoleGiveResponse PDU, the MRM provider generates an MRM-Room-Role-Give confirm primitive and sends it to the Control MRMSAP.

RoomRoleGiveIndication

The MRM Provider checks the result in the response primitive if it was successful; then, a RoomRoleGiveIndication is generated and sent to the MRM top provider on the MRM-Conference-Management-Channel at High data priority. (See Table 64.)

Table 64/T.137 – RoomRoleGiveIndication MRMPDU

Content	Source	Sink
Giving Node MRM User ID	Request	Indication
Recipient Node (MRM User ID)	Request	Indication
MRM Room Identifier	Request	Indication
MRM Room Role Identifier	Request	Indication
Group Identifier	Request	Indication

10.10.5 Changing role permissions

RolePermissionsModifyRequest

Service link: 9.8.2.1

On receipt of a MRM-Role-Permission-Modify primitive, the MRM provider constructs a RolePermissionsModifyRequest PDU from the request and inserts its node's MRM User ID; it then sends the PDU to the MRM top provider on the MRM-Conference-Management-Channel at High data priority. (See Table 65.)

Table 65/T.137 – RolePermissionsModifyRequest MRMPDU

Content	Source	Sink
Requesting Node MRM User ID	MRM Provider	MRM Top Provider
MRM Room Identifier	Request	MRM Top Provider
MRM Room Role Identifier	Request	MRM Top Provider
Permissions List	Request	MRM Top Provider

On receipt of a RolePermissionsModifyRequest PDU the top MRM provider checks whether the requesting node is authorized to modify role permissions by looking at its existing role permissions. If it is authorized, the MRM top provider sets the result to "Successful".

RolePermissionsModifyResponse

The MRM top provider constructs a RolePermissionsModifyResponse PDU and sends it to the requester's User ID at High data priority. (See Table 66.)

Table 66/T.137 – RolePermissionsModifyResponse MRMPDU

Content	Source	Sink
Result	Request	MRM Top Provider

On receipt of a RolePermissionsModifyResponse PDU, the MRM provider generates an MRM-Role-Permissions-Modify confirm primitive and issues it to the Control MRMSAP.

10.11 Permission PDUs

10.11.1 Asking the chair for permission to take action

PermissionAskRequest

Service link: 9.9.1.1

On receipt of an MRM-Permission-Ask request primitive, the MRM provider checks its local database in order to confirm that there is an active chair in the room. This is done by looking in its local database at the assigned role list. If there is no chair, then the request is rejected locally and the Confirm is generated and sent to the node controller on the Control MRMSAP.

If the chair is contained in the MRM provider's assigned role list, then the MRM User ID associated with the chair entry is used as the destination address for the PermissionAskRequest PDU. The PDU is generated and sent to the chair's MRM User ID at High data priority and the MRM-Permission-Ask primitive is generated with the result "success" and sent to the Control MRMSAP. (See Table 67.)

Table 67/T.137 – PermissionAskRequest MRMPDU

Content	Source	Sink
Requesting Node MRM User ID	Request	Indication
MRM Room Identifier	Request	Indication
Permission Requested	Request	Indication
Group Permission Request	Request	Indication
Associated Service	Request	Indication

On receipt of a PermissionAskRequest PDU, the MRM Provider checks whether it is the current "chair" role holder; if it is not, the request is ignored. Otherwise, it generates an MRM-Permission-Ask indication and sends it to its Control MRMSAP. The reply is returned using the Permission grant mechanism and is initiated by the node controller.

10.11.2 Relinquishing Permission

PermissionRelinquishRequest

Service link: 9.9.1.2

On receipt of an MRM-Permission-Relinquish request primitive, the MRM provider checks its local database to confirm whether there is an active chair in the room. This is done by looking in its local database at the assigned Role list. If there is no chair, then the request is rejected locally and a Confirm is generated containing the result "NoChair". The Confirm is sent to the node controller, on the Control MRMSAP.

If the chair is on the MRM provider's AssignedRole list, the PermissionRelinquishRequest PDU is generated and sent to the chair's MRM User ID, at High data priority and the MRM-Permission-Relinquish confirm primitive is generated with the result "success" and sent to the Control MRMSAP. (See Table 68.)

Table 68/T.137 – PermissionRelinquishRequest MRMPDU

Content	Source	Sink
Requesting Node MRM User ID	Request	Indication
Permission Relinquished	Request	Indication
Release Reason	Request	Indication
MRM Room Identifier	Request	Indication
Group Permission Release	Request	Indication
Associated Service	Request	Indication

On receipt of a PermissionRelinquishRequest PDU, the MRM Provider checks whether it is the current chair role holder. If it is not, it ignores the request. Otherwise, it generates an MRM-Permission-Relinquish indication and sends it to its Control MRMSAP.

10.11.3 Chair grants permission

PermissionGrantRequest

Service link: 9.9.1.3

On receipt of an MRM-Permission-Grant request primitive, the chair's MRM provider sends a PermissionGrantRequest PDU to the MRM User ID of the node identified in the request primitive at High data priority. The MRM provider then generates an MRM-Permission-Grant confirm and issues it to the Control MRMSAP. (See Table 69.)

Table 69/T.137 – PermissionGrantRequest MRMPDU

Content	Source	Sink
Request Handle	Request	Indication
Requesting Node	Request	Indication
Room Identifier	Request	Indication
Permission requested	Request	Indication
Group Permission requested	Request	Indication
Associated Service	Request	Indication
MRM Access Capability	Request	Indication
Result	Request	Indication

On receipt of a PermissionGrantRequest PDU, the MRM generates an MRM-Permission-Grant indication and sends it to its Control MRMSAP.

10.12 Source Identification PDUs

SourceIdentificationRequest

Service link: 9.10.1.1

On receipt of an MRM-Source-Identification-Request primitive, the MRM provider shall send a SourceIdentificationRequest PDU to the MRM-Conference-Management-Channel, specifying High data priority. (See Table 70.)

Table 70/T.137 – SourceIdentificationRequest MRMPDU

Content	Source	Sink
Source Handle	Request	Indication
Source Channel Identifier	Request	Indication
Media Type	Request	Indication
Room ID	Request	Indication
Stream Descriptor	Request	Indication
Reason Code	Request	Indication

On receipt of a SourceIdentificationRequest PDU, the MRM provider generates an MRM-Source-Identification indication and sends it to its Control MRMSAP.

10.13 Indication PDUs

10.13.1 Notification of privacy

PrivacyNotifyRequest

Service link: 9.10.1.2

On receipt of an MRM-Privacy-Notify request primitive, the MRM provider sends a PrivacyNotifyRequest PDU to the MRM-Conference-Management. Channel at High data priority. (See Table 71.)

Table 71/T.137 – PrivacyNotifyRequest MRMPDU

Content	Source	Sink
Source Handle	Request	Indication
Room ID	Request	Indication
Media channel ID	Request	MRM Provider
Media Type	Request	MRM Provider
Invoke Privacy Remotely	Request	MRM Provider
Privacy Status	Request	Indication

PrivacyNotifyResponse

On receipt of an MRM-Privacy-Notify response primitive, the MRM provider generates an MRM-Privacy-Notify indication primitive and issues it to the Control MRMSAP. Then, it updates its database to record the privacy mode of the requesting node and signals the privacy mode to any services that are active, such as Audio mixing, etc. Then, a PrivacyNotifyResponse PDU is sent on the MRM User ID of the requesting node at High data priority. (See Table 72.)

Table 72/T.137 – PrivacyNotifyResponse MRMPDU

Content	Source	Sink
Source Handle	MRM Provider	Confirm
Media Channel Identifier	MRM Provider	Confirm
Media Type	MRM Provider	Confirm
Privacy Status	MRM Provider	Confirm

On receipt of a PrivacyNotifyResponse PDU, the MRM provider generates an MRM-Privacy-Notify confirm and issues it to the Control MRMSAP.

OnAirIndicationRequest

Service link: 9.10.1.3

On receipt of an MRM-On-Air-Indication request primitive the MRM provider sends a Request PDU to the MRM User ID of the node identified in the request primitive at High data priority. The MRM provider then generates an MRM-confirm and issues it to the Control MRMSAP. (See Table 73.)

Table 73/T.137 – OnAirIndicationRequest MRMPDU

Content	Source	Sink
Source Handle	MRM Provider	Indication
Media Channel Identifier	Request	Indication
Room Identifier	Request	Indication
On Air Status Flag	Request	Indication

On receipt of an OnAirIndicationRequest PDU, the MRM provider generates an MRM-On-Air-indication and issues it to the Control MRMSAP.

10.14 Service Command PDUs

ServiceCommandRequest

Service link: 9.10.1.4

On receipt of an MRM-Service-Command request primitive, the MRM provider generates a ServiceCommandRequest PDU which is sent to the MRM Conference Management channel at Top data priority. (See Table 74.)

Table 74/T.137 – ServiceCommandRequest MRMPDU

Content	Source	Sink
Requesting Node	MRM Provider	Indication
MRM Room ID	Request	Indication
MRM Service Command	Request	Indication
Service Name ID	Request	Indication
Service Mode	Request	Indication
Participation Options	Request	Indication
Participant and Resources List	Request	Indication
Include Observer Nodes – Flag	Request	Indication

The MRM top provider shall, on receipt of the ServiceCommandRequest PDU, verify that the requesting node is authorized to execute the command specified in the MRM Service Command parameter by checking the conference framework role permissions list for the requesting node. The MRM top provider must also check the service specified in the MRM service is permitted by the conference framework. This is done by looking for the Service ID in the framework's permitted services list. If the request is permitted by the framework, the MRM top provider generates an MRM-Service-Command primitive and sends it to the Conference Manager on the Control MRMSAP.

ServiceCommandResponse

The Top MRM provider responds to the MRM-Service-Command-Response primitive by checking the result code; if it is "Successful", then the MRM top provider executes the MRM Service Command. The MRM Top provider generates a ServiceCommandResponse PDU which it sends to the requesting node with the result "Successful" if the Service command was accepted, or "prohibited by Framework" or "Service Unavailable" if the request was not accepted. (See Table 75.)

Table 75/T.137 – ServiceCommandResponse MRMPDU

Content	Source	Sink
MRM Room ID	Request	Indication
MRM Service Command	Request	Indication
Service Name ID	Request	Indication
Service Mode	Request	Indication
Result	Request	Indication

An MRM provider, on receipt of ServiceCommandResponse subsequent to sending an ServiceCommandRequest, shall generate an MRM-Service-Command confirm primitive containing the Result parameter indicating if the request was successful or not.

10.15 Audio Management PDUs

10.15.1 Service initialization

The Audio Bridging service has a defined behaviour for events that take place in an MRM room that impact upon the Audio Bridging service.

The conference framework specifies the requirement and the configuration for the Audio Bridging service. The service capabilities are derived from this configuration contained in the Framework's Service Constructor.

The Audio Bridging service is launched by the MRM top provider, in response to a framework requirement. If there are network elements participating in the conference, the launching of the service causes a ServiceCommand PDU containing the Command to launch. This PDU is sent to the MRM-Conference-Management-Channel at Top data priority. An Indication is also generated locally to the conference manager and issued to the Control MRMSAP. The MRM top provider then includes service capability within the Room roster.

When a node is subscribing to the Audio Bridging service it is:

- unsubscribing to the Audio Bridging service;
- entering an MRM room;
- leaving an MRM room.

10.15.2 Audio Mixer Setup

AudioMixerSetRequest

Service link: 9.12

On receipt of an MRM-Audio-Mixer-Set request primitive, the MRM provider generates an AudioMixerSetRequest PDU and sends it to the MRM top provider on the MRM-Conference-Management-Channel at Top data priority. (See Table 76.)

Table 76/T.137 – AudioMixerSetRequest MRMPDU

Content	Source	Sink
Originating Node MRM User ID	MRM Provider	Indication
MRM Room Identifier	Request	Indication
Audio Mixing Mode	Request	Indication
Add Node-Stream (List)	Request	Indication

Table 76/T.137 – AudioMixerSetRequest MRMPDU (concluded)

Content	Source	Sink
Enable Node-Stream (List)	Request	Indication
Mute Node-Stream (List)	Request	Indication
Include Node-Stream (List)	Request	Indication
Remove Node-Stream List	Request	Indication
Request Audio Level Adjust	Request	Indication

On receipt of an AudioMixerSetRequest PDU, the MRM top provider first checks the role privileges to determine if the originating node is authorized to control audio mixes. If it is not, the result is set to "NotAuthorized". Otherwise, the result is set to "successful". Other network elements receiving the request must also check the privileges in their local version of the framework if they are going to take action based upon this request. Authorized requests result in recipients generating an MRM-Audio-Mixer-Set indication and issuing it to its Control MRMSAP.

AudioMixerSetResponse

The MRM top provider is responsible for sending the response. It generates an AudioMixerSetResponse PDU containing the result code and sends it to the MRM User ID of the requester at Top data priority. (See Table 77.)

Table 77/T.137 – AudioMixerSetResponse MRMPDU

Content	Source	Sink
Originating Node MRM User ID	Indication	Confirm
Result	Response	Confirm

On receipt of an AudioMixerSetResponse PDU, the MRM provider generates an MRM-Audio-Mixer-Set confirm primitive and issues it to its Control MRMSAP.

10.15.3 Audio Mixer status reporting

AudioMixReportRequest

Service link: 9.12.1.2

On receipt of an MRM-AudioMix-Report request primitive, the MRM top provider shall construct an AudioMixReportRequest PDU, then send it to the nodes in that room requiring Audio Reports, specifying High data priority to the Recipient's MRM User IDs. (See Table 78.)

Table 78/T.137 – AudioMixReportRequest MRMPDU

Content	Source	Sink
MRM User ID	Request	Indication
MRM Room Identifier	Request	Indication
New Speaker MRM User ID	Request	Indication
New Speaker Microphone Device ID	Request	Indication
Previous Speaker MRM User ID	Request	Indication

Table 78/T.137 – AudioMixReportRequest MRMPDU (concluded)

Content	Source	Sink
Previous Speaker Microphone Device ID	Request	Indication
List of Nodes included in mix	Request	Indication
List of Active Speakers	Request	Indication

On receipt of AudioMixReportRequest PDU, the MRM provider shall generate an MRM-AudioMix-Report indication primitive and issue it to the Control MRMSAP.

10.15.4 Inviting Audio-Only participation

InviteAudioOnlyNodeRequest

Service link: 9.12.1.4

On receipt of an MRM-Invite-Audio-Only-Node request primitive, the MRM provider generates an InviteAudioOnlyNodeRequest PDU and sends it to the MRM top provider at Top data priority. (See Table 79.)

Table 79/T.137 – InviteAudioOnlyNodeRequest MRMPDU

Content	Source	Sink
DestinationRoom ID Handle	Request	Indication
Node Communication Address	Request	Indication
Room Access Parameter	Request	Indication
Reason for Invitation	Request	Indication
Entry Method	Request	Indication

On receipt of an InviteAudioOnlyNodeRequest PDU, the MRM provider shall generate an MRM-Invite-Audio-Only-Node indication primitive and issue it to the Control MRMSAP.

InviteAudioOnlyNodeResponse

The top MRM provider responds to the MRM-Invite-Audio-Only-Mode-Node Response generating an InviteAudioOnlyNodeResponse PDU and sending it to the requesting node with the result, which will be "Accepted" or "prohibited by Framework" or "Function Unavailable" if the request was not rejected. (See Table 80.)

Table 80/T.137 – InviteAudioOnlyNodeResponse MRMPDU

Content	Source	Sink
Result	Response	Confirm

An MRM Provider, on receipt of InviteAudioOnlyNodeResponse subsequent to sending an InviteAudioOnlyNodeRequest, shall generate an MRM-Invite-Audio-Only-Node confirm primitive containing the Result parameter indicating if the request was successful or not.

10.15.5 Ejecting an-Audio-Only participant

EjectAudioOnlyNodeRequest

Service link: 9.12.1.4

On receipt of an MRM-Eject-Audio-Only-Node request primitive, the MRM provider generates an EjectAudioOnlyNodeRequest PDU and sends it to the MRM top provider at High data priority. (See Table 81.)

Table 81/T.137 – EjectAudioOnlyNodeRequest MRMPDU

Content	Source	Sink
Room ID Handle	Request	Indication
Node Communication Address	Request	Indication
NonT120node User ID	Request	Indication

On receipt of EjectAudioOnlyNodeRequest PDU, the MRM provider shall generate an MRM-Eject-Audio-Only-Node indication primitive and issue it to the Control MRMSAP.

EjectAudioOnlyNodeResponse

The top MRM provider responds to the MRM-Eject-Audio-Only-Node Response by generating an EjectAudioOnlyNodeResponse PDU and sending it to the requesting node with the result, which will be "Accepted" or "prohibited by Framework" or "Function Unavailable" if the request was not rejected. (See Table 82.)

Table 82/T.137 – EjectAudioOnlyNodeResponse MRMPDU

Content	Source	Sink
Result	Response	Confirm

An MRM provider, on receipt of an EjectAudioOnlyNodeResponse, subsequent to sending an EjectAudioOnlyNodeRequest, shall generate a MRM-Eject-Audio-Only-Node confirm primitive containing the Result parameter indicating if the request was successful or not.

CallStatusIndication

Service link: 9.12.1.4

On receipt of an MRM-Call-Status request primitive, the MRM top provider generates an CallStatusIndication PDU and sends it to the MRM provider of the call initiator at High data priority. (See Table 83.)

Table 83/T.137 – CallStatusIndication MRMPDU

Content	Source	Sink
Room ID Handle	Request	Indication
NonT120node User ID	Request	Indication
Status	Request	Indication

On receipt of a CallStatusIndication PDU, the MRM provider shall generate an MRM-Call-Status indication primitive and issue it to the Control MRMSAP.

11 ASN.1 Definitions

-- <<<<<<<<<< PDU MESSAGE ELEMENTS >>>>>>>>>>

MRM-PROTOCOL DEFINITIONS AUTOMATIC TAGS ::= -- PART 1

BEGIN

IMPORTS --Imports from GCC

ChannelID,
StaticChannelID,
DynamicChannelID,
UserID,
TextString,
SimpleNumericString,
SimpleTextFirstCharacter,
SimpleTextLastCharacter,
SimpleTextString,
SimpleNumericString,
DiallingString,
ExtraDiallingString,
SessionKey,
H221NonStandardIdentifier,
SubAddressString,
Key,
Handle,
Password

FROM GCC-PROTOCOL;

-- <<<<<<<<< Begin New ASN.1 For MRM >>>>>>>>>>

RosterSequenceNum ::= INTEGER(1..65535)
ParticipantList ::= SET OF UserID
RoomTemplate ::= INTEGER(1..65535)
ServiceProviderID ::= INTEGER(1..65536)
SiteID ::= INTEGER(1..65535)
Template ::= INTEGER(1..65535)
UserServiceHandle ::= INTEGER(1001..65535)
UserRoleHandle ::= INTEGER(1001..65535)
GroupID ::= ChannelID
SourceChannelID ::= INTEGER(0..65535)
RoomKey ::= OCTET STRING(SIZE(0..24))

GCCApplicationCapabilities ::= SEQUENCE

{ nodeType BOOLEAN OPTIONAL,
participantLevel BOOLEAN OPTIONAL,
mediaServices SET OF ENUMERATED
{ pSTNAudio (0),
h323Audio (1),
h324Audio (2),
h323Video (3),
h324Video (4),
audioMixing (5),
videoSwitching (6),
videoStreaming (7),
shadowTopProvider (8),
...
}OPTIONAL,
...
}

MRMKey ::= CHOICE

```
{
    roomPassword Password,
    ...
}
```

MRMFunctions ::= ENUMERATED

```
{
    framework (0),
    roster (1),
    rooms (2),
    groups (3),
    services (4),
    sessions (5),
    permissions (6),
    sourceID (7),
    privacy (8),
    telephony (9),
    audioMix (10),
    ...
}
```

MRMServiceID ::= CHOICE

```
{
    standardService ENUMERATED
    {
        audioMixing (0),
        ...
    },
    userDefinedService UserServiceHandle,
    ...
}
```

MRMServiceRequirements ::= SEQUENCE

```
{
    mRMServiceID MRMServiceID,

    requirement ENUMERATED
    {
        offer (0), -- Advertise capability only
        launch (1), -- Launched the service
        activate (2), -- Service Launched and made active
        -- ready for nodes to subscribe to.
        prohibited (3), -- Explicitly prohibit a service
        ...
    },

    participationInstructions ENUMERATED
    {
        mustSubscribe (0), -- applied to all media capable participants
        maySubscribe (1), -- applies to all media capable participants
        ...
    },

    observerInstructions ENUMERATED
    {
        mustSubscribe (0), -- applied to all media capable Observers
        maySubscribe (1), -- applies to all media capable Observers
        subscribeProhibited (2),
        ...
    },
}
```

...
}

Permission ::= ENUMERATED

{
 toSpeak (0),
 toBeSeen (1),
 toUseAppShare (2),
 toUseWhiteboard (3),
 toEnterRoom (4),
 ...
}

T120UserApps ::= ENUMERATED

{
 chat (0),
 bFT (1),
 whiteboard (2),
 appSharing (3),
 ...
}

MRRole ::= CHOICE

{
 standardRole **ENUMERATED**
 {
 chair (0),
 operator (1),
 groupMember (2),
 ...
 },
 userDefinedRoles **UserRoleHandle**,
 ...
}

RoomReference ::= SEQUENCE

{
 roomHandle **INTEGER(0..65535)**,
 roomConstructorID **INTEGER(0..65535)**,
 roomParent **INTEGER(0..65535)**,
 roomName **SimpleTextString**,
 roomDescription **SimpleTextString**,
 ...
}

RoomManagementStyle ::= ENUMERATED

{
 looselyCoupled (0),
 informal (1),
 semiFormal (2),
 formal (3),
 ...
}

RoomManagementRegime ::= ENUMERATED

{
 noChair (0),
 chairAvailable (1),
 chairActive (2),
 ...
}

RoomType ::= ENUMERATED

```
{
    lobby           (0),
    meetingRoom     (1),
    presentationRoom (2),
    gathering        (3),
    personalRoom     (4),
    teamConf         (5),
    panelConf        (6),
    ...
}
```

RoomMode ::= ENUMERATED

```
{
    permanent (0),
    persistent (1),
    dynamic    (2),
    private    (3),
    ...
}
```

Media ::= SEQUENCE

```
{
    audioFlg      BOOLEAN,
    videoFlg      BOOLEAN,
    dataFlg       BOOLEAN,
    rTDataFlg     BOOLEAN,
    ...
}
```

RoomAccessModel ::= ENUMERATED

```
{
    open           (0),
    controlled     (1),
    byInvitation   (2),
    ...
}
```

RoomEntryRegime ::= ENUMERATED

```
{
    direct           (0),
    byInvitation     (1),
    viaLobby         (2),
    viaAutomatedReception (3),
    viaDoorKeeper    (4),
    ...
}
```

RoomExitRegime ::= ENUMERATED

```
{
    exitToPreviousRoom (0),
    exitToLobby        (1),
    exitConference      (2),
    ...
}
```

RoomState ::= ENUMERATED

```
{
    created      (0),
    ready        (1),
    active       (2),
    suspended    (3),
    destroyed    (4),
}
```



```

}
...
}
RoomConfig ::= SEQUENCE
{
    roomID                ChannelID,
    roomsFrameworkID     Handle OPTIONAL,
    roomMode              RoomMode OPTIONAL,
    roomManagementModel  RoomManagementRegime OPTIONAL,
    roomManagementStyle  RoomManagementStyle OPTIONAL,
    roomType              RoomType OPTIONAL,
    roomMedia             Media OPTIONAL,
    roomAccess            RoomAccessModel OPTIONAL,
    roomEntryRegime      RoomEntryRegime OPTIONAL,
    roomExitRegime       RoomExitRegime OPTIONAL,
    roomHierarchy         INTEGER(1..8) OPTIONAL,
    requiredRoles         SET OF MRMRole OPTIONAL,
    requiredServices     SET OF MRMServiceID OPTIONAL,
    roomTemplate          Template,
    ...
}

```

```

RoomStatus ::= SEQUENCE
{
    roomState             RoomState,
    numParticipants      INTEGER(0..16384),
    numObservers         INTEGER(0..16384),
    activeRoles          SET OF MRMRole OPTIONAL,
    activeServices       SET OF MRMServiceID OPTIONAL,
    activeApplications   SET OF T120UserApps OPTIONAL,
    activeGroups         SET OF GroupID OPTIONAL,
    ...
}

```

```

RoomTemplatesList ::= SET OF SEQUENCE
{
    roomTemplateID       Template,
    ...
}

```

```

ConfState ::= ENUMERATED
{
    booked      (0),
    ready      (1),
    active      (2),
    suspended  (3),
    completed  (4),
    cancelled  (5),
    ...
}

```

```

ConfModel ::= ENUMERATED
{
    undefined      (0),
    formalMeeting  (1),
    informalMeeting (2),
    multiroomConf  (3),
    broadcastConf  (4),
    collaboration  (5),
    gathering      (6),
    ...
}

```

```

ConfConfig ::= SEQUENCE
{
    confName           SimpleTextString OPTIONAL,
    confDescription  SimpleTextString OPTIONAL,
    confModel        ConfModel OPTIONAL,
    multisiteFlag    BOOLEAN OPTIONAL,
    primarySite      SiteID OPTIONAL,
    slaveSiteList    SET SIZE(1..128) OF SiteID OPTIONAL,
    maxConventionalNodes INTEGER(0..8192) OPTIONAL,
    maxAnonymousNodes INTEGER(0..16384) OPTIONAL,
    permittedServicesList SET OF MRMSERVICEID OPTIONAL,
    permittedRolesList SET OF MRMRole OPTIONAL,
    groupsPermittedFlag BOOLEAN OPTIONAL,
    lobbyRequiredFlag BOOLEAN OPTIONAL,
    maxRooms        INTEGER(1..256) OPTIONAL,
    numRoomsDefined  INTEGER(1..256) OPTIONAL,
    prohibitedPDUsList SET OF MRMPDU OPTIONAL,
    ...
}

```

```

ConfStatus ::= SEQUENCE
{
    numParticipants  INTEGER(0..16384),
    numObservers    INTEGER(0..16384),
    numRooms        INTEGER(1..127),
    roomDepth       INTEGER(1..15),
    roomList        SET OF ChannelID,
    ...
}

```

```

UserDetails ::= SEQUENCE
{
    mrmsUserHandle  UserID,
    name            SimpleTextString ,
    company         SimpleTextString OPTIONAL,
    address         SimpleTextString OPTIONAL,
    e-mail          SimpleTextString OPTIONAL,
    tel            DiallingString OPTIONAL,
    fax            DiallingString OPTIONAL,
    ...
}

```

```

ConfRosterReporting ::= SEQUENCE -- Set of flags telling the MRM Top Provider
                                   -- what to include in Conference Roster Reports

```

```

{
    configuration    BOOLEAN,
    status           BOOLEAN,
    participantList  BOOLEAN,
    participantDetails BOOLEAN,
    roomTemplates   BOOLEAN,
    allowRosterEnquiry BOOLEAN,
    ...
}

```

```

RoomRosterReporting ::= SEQUENCE -- Set of flags telling the MRM Top Provider
                                   -- what to include in Room Roster Reports

```

```

{
    configuration    BOOLEAN,
    status           BOOLEAN,
    participantList  BOOLEAN,
    participantDetails BOOLEAN,

```

```

    roomTemplates          BOOLEAN,
    ...
}

ServiceCommand ::= ENUMERATED
{
    launch      (0),
    activate    (1),
    de-activate (2),
    subscribe   (3),
    ...
}

ScopeOfRoster ::= ENUMERATED
{
    conference (0),
    room       (1),
    group      (2),
    individual (3),
    ...
}

-- <<<<<<<<<< PART 2 - PDU Messages >>>>>>>>>>>>>>
-- >>>>>>>>>>>>>> FRAMEWORK - protocol: 10.6

RoomConstructor ::= SEQUENCE
{
    roomConstructorID          INTEGER(0..65535),
    frameworkRoomID           INTEGER(1..65536) OPTIONAL, -- Used if constructor specific to room
    roomPurposeDescriptor     SimpleTextString OPTIONAL,
    roomType                   RoomType OPTIONAL,
    roomMode                   RoomMode OPTIONAL,
    roomState                  RoomState OPTIONAL,
    roomAppearance             INTEGER(0..65535) OPTIONAL, -- User defined Template
    roomParent                 INTEGER(0..65535) OPTIONAL, -- Room ID of parent room
    roomManagementRegime      RoomManagementRegime OPTIONAL,
    roomManagementStyle       RoomManagementStyle OPTIONAL,
    roomAccessModel            RoomAccessModel OPTIONAL,
    roomEntryRegime            RoomEntryRegime OPTIONAL,
    roomExitRegime             RoomExitRegime OPTIONAL,
    childRoomsFlag            BOOLEAN OPTIONAL,
    maxParticipants            INTEGER(1..8192) OPTIONAL,
    maxObservers                INTEGER(1..16384) OPTIONAL,
    participantAccessKey       Password OPTIONAL,
    observerAccessKey          Password OPTIONAL,
    groupsPermittedFlag        BOOLEAN OPTIONAL,
    chairRequiredFlag          BOOLEAN OPTIONAL,
    chairAccessKey             Password OPTIONAL,
    presenterRequiredFlag      BOOLEAN OPTIONAL,
    presenterAccessKey         BOOLEAN OPTIONAL,
    operatedRequiredFlag      BOOLEAN OPTIONAL,
    operatorAccessKey          Password OPTIONAL,
    groupsRequireList          SET OF SEQUENCE
    {
        groupID                UserID,
        groupAccessKey          Password,
        ...
    } OPTIONAL,
    audioOnlySupport           BOOLEAN OPTIONAL,
    floorControlRequiredFlag   BOOLEAN OPTIONAL,
    servicesRequirements       SET OF MRMSserviceRequirements OPTIONAL,
    ...
}

```

```

AudioMixingService ::= SEQUENCE
{
    serviceConstructorHandle      INTEGER(0..65535),
    audioMode ENUMERATED
    {
        mono          (0),
        stereo        (1),
        spatial        (2),
        ...
    } OPTIONAL,
    maxNumChannelsPerParticipant  INTEGER(1..8) OPTIONAL,
    maxNumParticipantsPerNode    INTEGER(1..512) OPTIONAL,
    numMixerPortsInConf          INTEGER(3..8192) OPTIONAL,
    numAudioInputsRequired       INTEGER(2..512) OPTIONAL,
    numInterBridgeChannels        INTEGER(0..128) OPTIONAL,
    startupMode ENUMERATED
    {
        default       (0),
        mixAll         (1),
        manual         (2),
        ...
    } OPTIONAL,

    mixingManagement      ENUMERATED
    {
        central          (0),
        distributed      (1),      -- Reserved for future use
        manual           (2),
        ...
    } OPTIONAL,
    supportedFeaturesList SEQUENCE
    {
        serviceAnnouncements  BOOLEAN,
        floorControl          BOOLEAN,
        stereo                 BOOLEAN,
        spatialAudio           BOOLEAN,
        surroundSound         BOOLEAN,
        ...
    } OPTIONAL,
    ...
}

```

```

FrameworkLoadIndication ::= SEQUENCE
{
    confFramework SEQUENCE
    {
        newDefaultsFlag      BOOLEAN, --Allows Defaults to be changed
        originatorId         Handle OPTIONAL,
        frameworkId          Handle OPTIONAL,
        confName              SimpleTextString OPTIONAL,
        confDescription       SimpleTextString OPTIONAL,
        confModel             ConfModel OPTIONAL,
        multisiteFlag        BOOLEAN OPTIONAL,
        maxConventionalNodes  INTEGER(0..8192) OPTIONAL,
        maxAnonymousNodes    INTEGER(0..16384) OPTIONAL,
        permittedServicesList SET OF MRMSERVICEID OPTIONAL,
        permittedRolesList   SET OF MRMRole OPTIONAL,
        groupsPermittedFlag  BOOLEAN OPTIONAL,
        synchDataConfFlag    BOOLEAN OPTIONAL,
        prohibitedPDUsList   SET OF MRMFunctions OPTIONAL,
        maxRooms              INTEGER(1..256) OPTIONAL,
        maxRoomDepth         INTEGER(1..8) OPTIONAL,
    }
}

```

```

        numRoomsDefined          INTEGER(1..256) OPTIONAL,
        partConfRosterReporting  BOOLEAN OPTIONAL,
        partRoomRosterReporting  BOOLEAN OPTIONAL,
        obsvrConfRosterReporting BOOLEAN OPTIONAL,
        obsvrRoomRosterReporting BOOLEAN OPTIONAL,
        roomList                 SET SIZE(0..255) OF RoomReference,
        roomConstructorList      SET SIZE(0..255) OF RoomConstructor,
        ...
    },
    ...
}

```

-- >>>>>>>>>> ROSTER - protocol link: 10.7

RosterReportIndication ::= SEQUENCE

```

{
    rosterSequenceNum    INTEGER (1..65535),
    scopeOfReport        ScopeOfRoster,
    confConfig            ConfConfig OPTIONAL,
    confStatus           ConfStatus OPTIONAL,
    confParticipantList  SET OF UserID OPTIONAL,
    roomConfig           RoomConfig OPTIONAL,
    roomStatus           RoomStatus OPTIONAL,
    roomParticipantsList SET OF UserID OPTIONAL,
    participantDetails   UserDetails OPTIONAL,
    ...
}

```

RosterUpdateIndication ::= SEQUENCE

```

{
    rosterSequenceNum    INTEGER (1..65535),
    confConfig            ConfConfig OPTIONAL,
    confStatus           ConfStatus OPTIONAL,
    confParticipantList  SET OF UserID OPTIONAL,
    roomConfig           RoomConfig OPTIONAL,
    roomStatus           RoomStatus OPTIONAL,
    roomParticipantsList SET OF UserID OPTIONAL,
    participantDetails   UserDetails OPTIONAL,
    ...
}

```

RosterEnquireIndication ::= SEQUENCE

```

{
    scopeOfEnquire      ScopeOfRoster,
    detailFlag          BOOLEAN,
    channel              ChannelID,
    configurationFlag   BOOLEAN,
    statusFlag          BOOLEAN,
    participantListFlag BOOLEAN,
    participantDetailsFlag BOOLEAN,
    roomTemplatesFlag   BOOLEAN,
    ...
}

```

-- >>>>>>>>>> ROOMS - protocol:10.8

RoomCreateRequest ::= SEQUENCE

```

{
    roomID                ChannelID OPTIONAL,
    roomOwnerFlag        BOOLEAN,
    roomParent            INTEGER(1..65535) OPTIONAL,
    frameworkRoomID     INTEGER(1..65535) OPTIONAL,
    roomTemplateFlag     BOOLEAN,
}

```

```

roomTemplateID          INTEGER(1..65535) OPTIONAL,

roomConfig SEQUENCE
{
    roomMode              RoomMode OPTIONAL,
    roomMedia            Media OPTIONAL,
    roomAccess          RoomAccessModel OPTIONAL,
    roomEntryRegime     RoomEntryRegime OPTIONAL,
    receptionRooms     SET OF ChannelID OPTIONAL,
    roomExitRegime     RoomExitRegime OPTIONAL,
    roomName            SimpleTextString OPTIONAL,
    roomPurpose        SimpleTextString OPTIONAL,
    numberOfParticipants INTEGER(0..16383) OPTIONAL,
    numberOfObservers  INTEGER(0..16383) OPTIONAL,

    requiredRolesList SEQUENCE
    {
        role MRMRole,
        ...
    } OPTIONAL,

    requiredServicesList SEQUENCE
    {
        service MRMServiceID,
        ...
    },
    ...
}

```

```

RoomCreateResponse ::= SEQUENCE
{
    roomID ChannelID,
    roomMasterKey RoomKey,
    result ENUMERATED
    {
        successful (0),
        roomDepthExceeded (1),
        maxRoomsExceeded (2),
        prohibitedByFramework (3),
        notAuthorized (4),
        invalidParameter (5),
        invalidParentID (6),
        outOfMCSResources (7),
        ...
    },
    ...
}

```

```

RoomEnterRequest ::= SEQUENCE
{
    enterRoomID ChannelID,
    currentRoomID ChannelID,
    suspendFlag BOOLEAN,
    participationType ENUMERATED
    {
        participant (0),
        observer (1),
        ...
    },
    roomAccessKey RoomKey OPTIONAL,
    reasonEntering SimpleTextString,
}

```

```

    ...
}

RoomEnterResponse ::= SEQUENCE
{
    destRoomID      ChannelID,
    roomAccessKey   RoomKey OPTIONAL,
    channelToJoin   ChannelID OPTIONAL,
    reasonEntering   SimpleTextString,

    result ENUMERATED
    {
        successful          (0),
        privateRoom        (1),
        gotoSpecifiedRoom  (2),
        invalidRoomID      (3),
        roomLimitsExceeded (4),
        notAuthorized       (5),
        ...
    },
    ...
}

```

```

RoomInviteIndication ::= SEQUENCE
{
    roomID           ChannelID, -- Also used as coms channel for room
    roomKey          MRMKey,
    invitedNodeList  SET OF UserID,
    invitingNodeID   UserID,
    reasonstring     SimpleTextString,
    ...
}

```

-- >>>>>>>>>> GROUPS - protocol: 10.8.4

```

GroupCreateRequest ::= SEQUENCE
{
    groupOwner       UserID,
    groupScopeFlag   BOOLEAN,
    roomID           ChannelID OPTIONAL,
    groupName        SimpleTextString,
    groupFunction     SimpleTextString,
    orderFlag        BOOLEAN,
    memberList       SET OF UserID OPTIONAL,
    ...
}

```

```

GroupCreateResponse ::= SEQUENCE
{
    roomID           ChannelID,
    groupHandle      ChannelID,
    result           ENUMERATED
    {
        success        (0),
        failed         (1),
        ...
    },
    ...
}

```

```

GroupDisbandRequest ::= SEQUENCE
{
    groupOwner      UserID,
    roomID         ChannelID,
    groupHandle    ChannelID,
    ...
}

```

```

GroupDisbandResponse ::= SEQUENCE
{
    groupHandle    ChannelID,
    result         ENUMERATED
    {
        successful      (0),
        notGroupOwner  (1),
        invalidGroupID (2),
        notAuthorized  (3),
        ...
    },
    ...
}

```

```

GroupMemberSpecifyRequest ::= SEQUENCE
{
    groupIdentifier      UserID,
    addMemberList       SET OF UserID,
    removememberList    SET OF UserID,
    specifyMemberList   SET OF UserID,
    ...
}

```

```

GroupMemberSpecifyResponse ::= SEQUENCE
{
    groupIdentifier      ChannelID,
    result ENUMERATED
    {
        successful      (0),
        notAuthorized  (1),
        invalidGroupID (2),
        ...
    },
    ...
}

```

-- >>>>>>>>>> *SESSION ASSOCIATION - protocol: 10.9*

```

SessionAssociateRequest ::= SEQUENCE
{
    roomId          ChannelID,
    associatedDataSessions SET OF SessionKey,
    ...
}

```

```

SessionAssociateResponse ::= SEQUENCE
{
    result          ENUMERATED
    {
        successful      (0),
        notAuthorized  (1),
        ...
    },
}

```



```

    ...
}
-- >>>>>>>>>> ROLES - protocol:9.9.2.1

```

```

RoomRoleAssignRequest ::= SEQUENCE
{
    requesterUserID      UserID,
    roleHolderUserID     UserID,
    roomID               ChannelID,
    roomRole             MRMRole,
    roleLabel            SimpleTextString,
    roleAccessKey       MRMKey,
    groupID              GroupID,
    ...
}

```

```

RoomRoleAssignResponse ::= SEQUENCE
{
    roomID               ChannelID,
    roomRole             MRMRole,
    result               ENUMERATED
    {
        successful          (0),
        roleNotPermitted   (1),
        roleAlreadyAssigned (2),
        invalidRoleKey     (3),
        ...
    },
    ...
}

```

```

RoomRoleReleaseRequest ::= SEQUENCE
{
    roomID               ChannelID,
    roleHolderID         UserID,
    role                 MRMRole,
    groupID              ChannelID,
    releaseReason        SimpleTextString,
    ...
}

```

```

RoomRoleReleaseResponse ::= SEQUENCE
{
    roomID               ChannelID,
    roomRole             MRMRole,
    result               ENUMERATED
    {
        successful          (0),
        notRoleHolder      (1),
        invalidRoom        (2),
        invalidGroup       (3),
        ...
    },
    ...
}

```

```

RoomRoleReleaseIndication ::= SEQUENCE
{
    roomID               ChannelID,
    roleHolderID         UserID,
    role                 MRMRole,
    groupID              ChannelID,
    releaseReason        SimpleTextString,

```

```

    ...
}

RoomRolePleaseRequest ::= SEQUENCE
{
    requesterID      UserID,
    roomID           ChannelID,
    roomRole        MRMRole,
    groupID         ChannelID,
    ...
}

```

```

RoomRoleGiveRequest ::= SEQUENCE
{
    roleHolderID    ChannelID,
    recipientID     ChannelID,
    roomID          ChannelID,
    roomRole        MRMRole,
    groupID         ChannelID,
    ...
}

```

```

RoomRoleGiveResponse ::= SEQUENCE
{
    roomID          ChannelID,
    roomRole        MRMRole,
    groupID         ChannelID,
    result          ENUMERATED
    {
        successful      (0),
        notRoleHolder   (1),
        ...
    },
    ...
}

```

```

RoomRoleGiveIndication ::= SEQUENCE
{
    roleGiver       UserID,
    roleRecipient   UserID,
    roomID          ChannelID,
    roomRole        MRMRole,
    groupID         ChannelID,
    ...
}

```

```

RolePermissionsModifyRequest ::= SEQUENCE
{
    requestingNodeID      UserID,
    createRole            BOOLEAN,
    passRole              BOOLEAN,
    askforRoleFlag       BOOLEAN,
    modifyRolePermissions  BOOLEAN,
    modifyotherRoles     BOOLEAN,
    roomCreationRights   BOOLEAN,
    roomInheritance      BOOLEAN,
    groupModification    BOOLEAN,
    groupOrdering        BOOLEAN,
    requestPermission    BOOLEAN,
    operateFloorControl  BOOLEAN,
    inviteEjectParticipants  BOOLEAN,
}

```

```

viewDataApps          BOOLEAN,
controlDataApps       BOOLEAN,
viewSpecificApp       SessionKey,
controlSpecificApp    SessionKey,
controlAudioMix       BOOLEAN,
controlAudioDevices   BOOLEAN,
inviteAudioParticipants BOOLEAN,
mutable               BOOLEAN,
controlVideoDevices   BOOLEAN,
...
}

```

RolePermissionsModifyResponse ::= SEQUENCE

```

{
  result ENUMERATED
  {
    success      (0),
    failed       (1),
    ...
  },
  ...
}

```

-- >>>>>>>>> PERMISSIONS - protocol: 10.11

PermissionAskRequest ::= SEQUENCE

```

{
  requestingID      UserID,
  roomID            ChannelID,
  releaseReason     SimpleTextString,
  permission        Permission,
  privateRoomID    ChannelID, -- Room in which to process request
  groupRequest      ChannelID,
  associatedService MRMSERVICEID,
  ...
}

```

PermissionRelinquishRequest ::= SEQUENCE

```

{
  requestingID      UserID,
  releaseReason     SimpleTextString,
  permission        Permission,
  roomID            ChannelID,
  groupPermission   BOOLEAN,
  associatedService MRMSERVICEID,
  ...
}

```

PermissionGrantRequest ::= SEQUENCE

```

{
  roomID            ChannelID,
  permissionGranted Permission,
  groupGrant        BOOLEAN,
  associatedService MRMSERVICEID,
  accessCap         ENUMERATED
  {
    shared          (0),
    exclusive       (1),
    ...
  },
  ...
}

```

-- >>>>>>>>> *SOURCE IDENTIFICATION - protocol: 10.12*

SourceHandle ::= SEQUENCE

```
{
    mRMChannel      UserID,
    mediaPipeTag    INTEGER(0..1023),
    ...
}
```

-- MRM User ID of node providing source
-- extension tag to distinguish
-- between different sources

SourceIdentificationRequest ::= SEQUENCE

```
{
    sourceHandle    SourceHandle,
    sourceChannelId ChannelID,
    mediatype       Media,
    roomID          ChannelID,
    streamDescriptor SimpleTextString,
    reasonCode      SimpleTextString,
    ...
}
```

-- >>>>>>>>> *ON-AIR INDICATION - protocol: 10.13*

OnAirIndicationRequest ::= SEQUENCE

```
{
    mediaChannelID ChannelID,
    sourceHandle    SourceHandle,
    roomID          ChannelID,
    onAirFlag       BOOLEAN,
    ...
}
```

-- >>>>>>>>> *PRIVACY - protocol: 10.13.1*

PrivacyNotifyRequest ::= SEQUENCE

```
{
    sourceHandle    SourceHandle,
    roomID          ChannelID,
    mediaChannelId ChannelID,
    mediatype       Media,
    remotePrivacy   BOOLEAN,
    privacyStatus   BOOLEAN,
    ...
}
```

PrivacyNotifyResponse ::= SEQUENCE

```
{
    sourceHandle    UserID,
    roomID          ChannelID,
    mediaChannel    ChannelID OPTIONAL,
    privacyStatus   BOOLEAN,
    ...
}
```

-- >>>>>>>>> *SERVICE COMMAND - protocol: 10.14*

ServiceCommandRequest ::= SEQUENCE

```
{
    requestingNode UserID,
    roomID         UserID,
}
```

```

command      ENUMERATED
{
    launch      (0),
    subscribe   (1),
    unSubscribe (2),
    terminate   (3),
    ...
},

service      ENUMERATED
{
    audioMix    (0),
    ...
},

mode         ENUMERATED
{
    default     (0),
    ...
},
...
}

```

```

ServiceCommandResponse ::= SEQUENCE
{
    requestingNode  UserID,
    result           ENUMERATED
    {
        successful      (0),
        notPermitted    (1),
        ...
    },
    ...
}

```

-- >>>>>>>>>> *AUDIO MIXER - protocol: 10.15*

```

AudioMixReportRequest ::= SEQUENCE
{
    userID           UserID,
    roomID           ChannelID,
    newSpkrUsr      UserID,
    newspkrDev      Handle,
    prevSpkrUsr     UserID,
    prevSpkrDev     Handle,
    nodelist        SEQUENCE OF UserID,
    activeSpkrList SEQUENCE OF UserID,
    ...
}

```

```

AudioMixSetRequest ::= SEQUENCE
{
    requester       UserID,
    roomID           ChannelID,
    mixingMode      ENUMERATED
    {
        default       (0),
        ...
    },
    addAudioStreamList SET OF Handle,
    enableAudioStreamList SET OF Handle,
    muteAudioStreamList SET OF Handle,

```

```

    removeAudioStreamList SET OF Handle,
    audioLevelAdjust      SEQUENCE
{
    scope          ChannelID,
    abslevel       INTEGER(0..100) OPTIONAL,
    uplevel        INTEGER(0..100) OPTIONAL,
    downlevel      INTEGER(0..100) OPTIONAL,
    ...
},
...
}

```

AudioMixSetResponse ::= SEQUENCE

```

{
    result ENUMERATED
    {
        success      (0),
        ...
    },
    ...
}

```

InviteAudioOnlyNodeRequest ::= SEQUENCE

```

{
    roomID          ChannelID,
    communicationAddress CHOICE
    {
        phoneNum      SEQUENCE
        {
            countryCode DiallingString,
            number       DiallingString,
            extension    DiallingString,
            ...
        },
        ipAddress      SubAddressString,
        ...
    },
    roomAccessKey   RoomKey,
    invitationReason SimpleTextString,
    entryMethod     ENUMERATED
    {
        direct        (0),
        viaSideConf   (1),
        ...
    },
    ...
}

```

InviteAudioOnlyNodeResponse ::= SEQUENCE

```

{
    result          ENUMERATED
    {
        success      (0),
        noReply      (1),
        failed       (2),
        ...
    },
    ...
}

```

```

EjectAudioOnlyNodeRequest ::= SEQUENCE
{
    roomID          ChannelID,
    communicationAddress CHOICE
    {
        phoneNum SET OF SEQUENCE
        {
            countryCode    DiallingString,
            number          DiallingString,
            extension       DiallingString
        },
        dialstring         ExtraDiallingString,
        iPAddress          SubAddressString,
        ...
    },
    nont120UserID       UserID,
    ...
}

```

```

EjectAudioOnlyNodeResponse ::= SEQUENCE
{
    result          ENUMERATED
    {
        success      (0),
        invalidUserID (1),
        notAuthorized (2),
        ...
    }
}

```

```

CallStatusIndication ::= SEQUENCE
{
    roomID          ChannelID,
    nont120UserID   UserID,
    status          ENUMERATED
    {
        dialling      (0),
        engaged       (1),
        nU             (2),
        noReply       (3),
        answered      (4),
        mixed         (5),
        ...
    }
}

```

-- <<<<<<<<< PART 3 - Messages Sent using MCS-Send-Data >>>>>>>>>>

```

MRMPDU ::= CHOICE
{
    request    RequestPDU,
    response   ResponsePDU,
    indication IndicationPDU,
    ...
}

```

```

RequestPDU ::= CHOICE
{
    roomCreateRequest           RoomCreateRequest,
    roomEnterRequest           RoomEnterRequest,
    sessionAssociateRequest     SessionAssociateRequest,
    groupCreateRequest         GroupCreateRequest,
    groupDisbandRequest        GroupDisbandRequest,
    groupMemberSpecifyRequest   GroupMemberSpecifyRequest,
    serviceCommandRequest      ServiceCommandRequest,
    roomRoleAssignRequest      RoomRoleAssignRequest,
    roomRoleReleaseRequest     RoomRoleReleaseRequest,
    roomRolePleaseRequest       RoomRolePleaseRequest,
    roomRoleGiveRequest        RoomRoleGiveRequest,
    rolePermissionsModifyRequest RolePermissionsModifyRequest,
    permissionAskRequest       PermissionAskRequest,
    permissionRelinquishRequest PermissionRelinquishRequest,
    permissionGrantRequest     PermissionGrantRequest,
    sourceIdentificationRequest SourceIdentificationRequest,
    onAirIndicationRequest     OnAirIndicationRequest,
    privacyNotifyRequest       PrivacyNotifyRequest,
    audioMixReportRequest      AudioMixReportRequest,
    audioMixSetRequest         AudioMixSetRequest,
    inviteAudioOnlyNodeRequest InviteAudioOnlyNodeRequest,
    ejectAudioOnlyNodeRequest  EjectAudioOnlyNodeRequest,
    ...
}

```

```

ResponsePDU ::= CHOICE
{
    roomCreateResponse         RoomCreateResponse,
    roomEnterResponse         RoomEnterResponse,
    sessionAssociateResponse   SessionAssociateResponse,
    groupCreateResponse       GroupCreateResponse,
    groupDisbandResponse      GroupDisbandResponse,
    groupMemberSpecifyResponse GroupMemberSpecifyResponse,
    serviceCommandResponse    ServiceCommandResponse,
    roomRoleAssignResponse    RoomRoleAssignResponse,
    roomRoleReleaseResponse   RoomRoleReleaseResponse,
    roomRoleGiveResponse      RoomRoleGiveResponse,
    rolePermissionsModifyResponse RolePermissionsModifyResponse,
    privacyNotifyResponse     PrivacyNotifyResponse,
    audioMixSetResponse       AudioMixSetResponse,
    inviteAudioOnlyNodeResponse InviteAudioOnlyNodeResponse,
    ejectAudioOnlyNodeResponse EjectAudioOnlyNodeResponse,
    ...
}

```

```

IndicationPDU ::= CHOICE
{
    frameworkLoadIndication    FrameworkLoadIndication,
    rosterReportIndication     RosterReportIndication,
    rosterUpdateIndication     RosterUpdateIndication,
    rosterEnquireIndication    RosterEnquireIndication,
    roomInviteIndication       RoomInviteIndication,
    roomRoleGiveIndication     RoomRoleGiveIndication,
    callStatusIndication       CallStatusIndication,
    ...
}

```

```

END -- <<<<<<<<< END ASN.1 >>>>>>>>>>

```


12 Requirements for running MRM

12.1 Summary of GCC services required by MRM

MRM assumes the services listed in the GCC primitives in Table 84. Primitives endorsed with an "M" are mandatory, those marked with an "O" are optional.

Table 84/T.137 – GCC primitives supported by MRM

GCC Primitive	Use
GCC-Application-Permission-To-Enrol indication	M
GCC-Application-Enrol request	M
GCC-Application-Enrol confirm	M
GCC-Application-Roster-Report indication	M

NOTE – MRM Terminals should ignore MRM Application capabilities for network elements; they are not needed by them.

12.2 Summary of use of the Multipoint Communication Service

All MRM communication shall be through the Multipoint Communication Service (MCS) as specified in Recommendation T.122. This subclause details how MRM makes use of MCS services, channel allocation, token allocation and data priorities.

12.2.1 MCS services

MRM assumes the MCS services indicated in Table 85. All primitives and parameters marked with an "M" are used by mandatory components of MRM. Items marked with an "O" are optional, and not required for MRM operation.

Table 85/T.137 – MCS services used by MRM

Primitives	Terminal use	Network Element use
MCS-Attach-User request	M	M
MCS-Attach-User confirm	M	M
MCS-Detach-User request	M	M
MCS-Detach-User indication	M	M
MCS-Channel-Join request	M	M
MCS-Channel-Join confirm	M	M
MCS-Channel-Leave request	M	M
MCS-Channel-Leave indication	M	M
MCS-Channel-Convvene request	O	M
MCS-Channel-Convvene confirm	O	M
MCS-Channel-Disband request	–	M
MCS-Channel-Disband indication	–	M
MCS-Channel-Admit request	–	M
MCS-Channel-Admit indication	–	M

Table 85/T.137 – MCS services used by MRM (concluded)

Primitives	Terminal use	Network Element use
MCS-Channel-Expel request	–	M
MCS-Channel-Expel indication	–	M
MCS-Send-Data request	M	M
MCS-Send-Data indication	M	M
MCS-Uniform-Send-Data request	O	O
MCS-Uniform-Send-Data indication	O	O
MCS-Token-Grab request	–	M
MCS-Token-Grab confirm	–	M
MCS-Token-Release request	–	M
MCS-Token-Release confirm	–	M

12.2.2 MCS channels used by MRM

See Table 86.

Table 86/T.137 – MCS channel assignments for MRM

Channel ID	Type	Mnemonic	Description
MRM-Channel-1	Static	MRM-Conference-Management-Channel	Joined by the MRM top provider; it may also be joined by other nodes that wish to shadow the top provider. Used by terminals to communicate with the MRM top provider.
MRM-Channel-2	Static	MRM-Conference-Communication-Channel	Joined by all terminals that take part in the MRM Conference. Used by the MRM top provider to distribute the MRM Conference Roster.
MRM-Channel-3	Static	MRM-Default-Room-Participant Channel	The Default Room channel used as the identifier for the default room and joined by MRM Participants.
MRM-Channel-4	Static	MRM-Default-Room-Observer Channel	
	Dynamic	MRM-Room-Participant-Channel	Used as the Room Identifier and joined by MRM participants.
	Dynamic	MRM-Room-Observer Channel	The channel that is allocated when an MRM Room is to accommodate MRM observers. It should be joined by Observers entering the room instead of the MRM-Room-Participant-Channel.
	Dynamic	MRM-Group-Membership Channel	This channel is allocated when an MRM Group is created it is used as the group identifier.

12.2.3 MCS Tokens used by MRM

See Table 87.

Table 87/T.137 – MCS token assignments for MRM

Token ID	Type	Mnemonic	Description
MRM-Token 0	Static	MRM-Top-provider-Token	MCS Token – Optional

12.2.4 Encoding of PDUs in MCS primitives

All PDUs defined in this Recommendation are encoded and placed in the data field of one of the MCS-Send-Data primitives. The bit string that results from the ASN.1 encoding is placed in the OCTET STRING used by MCS in the order such that for each octet the leading bit is placed in the most significant bit position, and the trailing bit is placed in the least significant bit position.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems