



Rui Carlos Ávila Leal

**Plataformas Broadcast/Multicast para Redes Móveis
(Mobile TV)**

Broadcast/Multicast Platforms for Mobile Networks (Mobile TV)



Rui Carlos Ávila Leal

**Plataformas Broadcast/Multicast para Redes Móveis
(Mobile TV)**

Broadcast/Multicast Platforms for Mobile Networks (Mobile TV)

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Dr. José Carlos da Silva Neves, Professor Catedrático da Universidade de Aveiro

o júri

presidente

Doutor Rui Jorge Morais Tomaz Valadas
Professor Associado com Agregação da Universidade de Aveiro

vogais

Doutor José Carlos da Silva Neves (Orientador)
Professor Catedrático da Universidade de Aveiro

Doutor Américo Manuel Carapeto Correia
Professor Catedrático do Departamento de Ciências e Tecnologia da Informação do Instituto Superior de Ciências do Trabalho e da Empresa

agradecimentos

Agradeço às pessoas que estiveram directamente ligadas a esta dissertação: Prof. José Carlos Neves, Prof. Manuel Dinis e Filipe Cabral Pinto. Um muito obrigado.

Aproveito também para agradecer à minha família pelo apoio que me foi demonstrado.

palavras-chave

UMTS, MBMS, Protocolo Diameter, Gmb, BM-SC, IMS, integração IMS-MBMS

resumo

O MBMS (Multimedia Broadcast/Multicast Services), uma evolução das redes UMTS, é um sistema que permite o envio de conteúdo via broadcast e multicast através redes móveis, como por exemplo a video-conferência, partilha de ficheiros e o streaming de áudio e vídeo (Mobile TV). Este envio de informação é do tipo ponto-multiponto, ou seja, existe apenas uma fonte que envia a informação para múltiplos receptores, ao invés do que acontece com as ligações ponto-a-ponto. Consequentemente, o uso dos recursos da rede é otimizado.

Com o MBMS surge uma nova entidade denominada por BM-SC (Broadcast Multicast Service Center) que funciona como uma interface entre os fornecedores de conteúdo e o resto da rede móvel. Esta entidade contém informação relacionada com os serviços disponibilizados pelo operador, os subscritores dos mesmos e desempenha funcionalidades como a autenticação, autorização, facturação, entre outras.

O BM-SC está directamente ligado ao GGSN (Gateway GPRS Support Node) através das interfaces Gi (dados) e Gmb (sinalização). A sinalização entre estes dois nós é feita a partir da troca de mensagens AAA (autenticação, autorização e *accounting*) baseadas no protocolo Diameter.

Assim, numa primeira fase, e para além da arquitectura MBMS, esta dissertação aborda o principal objectivo deste trabalho, ou seja, todo o procedimento realizado tendo em vista a implementação da sinalização entre o BM-SC e o GGSN através de uma implementação do protocolo Diameter já existente.

Este procedimento engloba inicialmente o estudo do protocolo Diameter, através de uma análise à sua arquitectura e à constituição das mensagens que o mesmo disponibiliza. Após a identificação de todos os procedimentos de sinalização entre o GGSN e o BM-SC, são abordados os passos que conduziram à implementação da interface Gmb através de uma aplicação em linguagem Java baseada numa máquina de estados. Assim, resulta daí uma aplicação onde eventos como o envio ou recepção de mensagens Diameter implicam acções específicas como o processamento da autorização de um utilizador para determinado serviço MBMS.

Numa segunda fase, esta dissertação contempla a evolução que hoje em dia se tem vindo a verificar no MBMS através da integração com outras tecnologias. A necessidade desta evolução prende-se com a crescente competição que existe entre o MBMS e outras tecnologias similares, como é o caso do DVB-H, DMB e do MediaFLO, e fundamentalmente pela necessidade de obter uma visão do MBMS para além do 3G, assegurando assim uma maior longevidade da tecnologia.

Uma dessas integrações, o IMS-MBMS proposto pelo projecto C-MOBILE e ao qual eu estive ligado, obtém nesta dissertação um maior destaque. Deste modo são abordadas as motivações que estiveram por trás da convergência do IMS (IP Multimedia Subsystem) com o MBMS, a arquitectura que surge desta convergência e a nova sinalização entre as entidades que resulta da distribuição das funcionalidades do BM-SC pela arquitectura integrada.

keywords

UMTS, MBMS, Diameter protocol, Gmb, BM-SC, IMS, IMS-MBMS Integration

abstract

MBMS (Multimedia Broadcast/Multicast Services), an enhancement of the UMTS networks, is a system capable of sending both broadcast and multicast content in mobile networks, such as video-conference, file sharing and audio/video streaming (Mobile TV). This data transmission is point-to-multipoint, i.e., there is only one source sending information to multiple receivers, contrary to point-to-point connections. Therefore, the use of network resources is optimized.

Along with MBMS comes a new entity called BM-SC (Broadcast Multicast Service Center) that represents an interface between the content providers and the mobile network. This entity contains all the information related to the available services, its subscribers and functionalities such as authentication, authorization, billing, among others.

The BM-SC is directly connected to GGSN (Gateway GPRS Support Node) through Gi (data) and Gmb (signalling) interfaces. The signaling process between these two nodes is done through the exchange of AAA (authentication, authorization and accounting) messages based on the Diameter protocol.

Therefore, in a first phase, besides the MBMS architecture, this document focuses in the main objective of this work, i.e., the whole process taken to implement the signaling between the BM-SC and the SGSN using an already existing Diameter protocol implementation.

This process initially involves the study of the Diameter Protocol, through the analysis of both its architecture and messages. After the identification of all signalling procedures that occur between the GGSN and the BM-SC, the steps that lead to the Gmb interface implementation through a Java application based on a state machine are addressed. Thus, it results an application where events like the sending or reception of Diameter messages involve specific actions like a user authorization procedure for a particular MBMS service.

In a second phase, this document describes the evolution that is happening nowadays in MBMS through the integration of other technologies. The need to evolve is due to the growing competition between MBMS and other similar technologies such as DVB-H, DMB and MediaFLO, but mainly because of the need to get a MBMS vision beyond 3G, thus ensuring greater longevity of the technology.

One of these integrations, the IMS-MBMS proposal by the C-MOBILE project (on which I was involved), obtains a particular focus on this document. Thus, the motivations behind the IMS (IP Multimedia Subsystem) and the MBMS convergence, the resulting architecture and the new signaling process between the entities after the redistribution of the BM-SC functionalities among the integrated architecture are dealt with.

Abbreviations

3GPP	3 rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
ABNF	Augmented Backus-Naur form
AS	Application Server
AVP	Attribute Value Pairs
BM-SC	Broadcast/Multicast Service Centre
BM-SE	Broadcast/Multicast Service Enabler
CSCF	Call Session Control Function
DMB	Digital Multimedia Broadcasting
DVB	Digital Video Broadcasting
DVB-H	Digital Video Broadcasting - Handheld
FLO (MediaFLO)	Forward Link Only
GGSN	Gateway GPRS Support Node
GM-SE	Group Management Service Enabler
GPRS	General Packet Radio System
GSM	Global System for Mobile Communications
HSDPA	High-Speed Downlink Packet Access
HSS	Home Subscriber Server
IMS	IP Multimedia Subsystem
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
I-CSCF	Interrogating – Call Session Control Function
LTE	Long Term Evolution
MBMS	Multimedia Broadcast/Multicast Services
MDF	Media Delivery Function
MDFC	Media Delivery Function Controller
MIMO	Multiple Input Multiple Output
OMA	Open Mobile Alliance
PDF	Policy Detection Function
PCRF	Policy and Charging Rules Function

P-CSCF	Proxy – Call Session Control Function
PSTN	Public Switch Telephone Network
PLMN	Public Land Mobile Network
RAN	Radio Access Network
RNC	Radio Network Controller
SAE	Service Architecture Evolution
S-CSCF	Serving – Call Session Control Function
SDP	Session Description Protocol
SEP	Service Enabler Plane
SFN	Single Frequency Network
SGSN	Serving GPRS Support Node
SIP	Session Initiation Protocol
SME	Session Management Enabler
SPR	Subscription Profile Repository
TCP	Transmission Control Protocol
UE	User Equipment
UMTS	Mobile Telecommunications System
XML	eXtensible Markup Language

Table of Contents

O jurí	2
Agradecimentos.....	3
Resumo.....	4
Abstract	5
Abbreviations.....	6
List of Figures.....	11
List of Tables.....	13
1. Introduction.....	14
2. MBMS Architecture.....	17
2.1 Introduction	17
2.2 MBMS Bearer Service and User Service.....	17
2.3 Broadcast and Multicast Mode	18
2.4 MBMS Architecture Model	20
2.5 GGSN	21
2.6 BM-SC	21
2.7 MBMS Attributes and Parameters	22
2.7.1 MBMS Bearer Context.....	22
2.7.2 MBMS UE Context.....	24
2.8 Gmb Interface	25
3. The Diameter Protocol	26
3.1 Description.....	26
3.2 History.....	26
3.3 Advantages of Diameter over RADIUS.....	26
3.4 The Diameter Architecture.....	27
3.5 Diameter Nodes.....	27
3.6 Diameter Messages.....	28
3.7 The OpenBloX™ Implementation	30
3.7.1 Overview	30
3.7.2 Reasons for its use on this work	30
3.7.3 OpenBloX™ Architecture.....	31
4. Diameter on Gmb.....	32
4.1 MBMS Procedures on Gmb	32

4.1.1 MBMS User Authorization.....	32
4.1.2 MBMS service registration/de-registration	32
4.1.3 MBMS session start/stop	32
4.1.4 MBMS User deactivation	33
4.2 Gmb Message Flows	33
4.2.1 Service Authorization Procedure	33
4.2.2. Session Start Procedure	35
4.2.3 Session Stop Procedure	35
4.2.4 Registration Procedure	36
4.2.5 De-registration Procedure (GGSN Initiated)	37
4.2.6 De-registration Procedure (BM-SC initiated)	37
4.2.7 Service Deactivation Procedure.....	38
4.2.7.1 Service Deactivation Procedure (BM-SC initiated)	40
4.2.8 Trace Session Activation Procedure.....	40
4.2.9 Trace Session Deactivation	41
4.2.10 UE Context Modification	42
4.3 Gmb Messages.....	42
4.3.1 AAR Command.....	43
4.3.2 AAA Command	43
4.3.3 STR Command	44
4.3.4 STA Command	44
4.3.5 RAR Command.....	45
4.3.6 RAA Command.....	46
4.3.7 ASR Command.....	46
4.3.8 ASA Command	46
5. Gmb interface implementation	48
5.1 Client/Server VS Peer-to-Peer.....	48
5.2 Developing the Server and Client applications	48
5.3 The concept behind State Machines	51
5.4 Managing events	51
5.5 Handling events with actions	54
5.6 The implementation interface	58
6. Running the Implementation	60
6.1 Establishing connection	60

6.2 Service Activation procedure	61
6.3 Registration procedure	64
6.4 Session Start procedure	66
6.5 Session Stop procedure	66
6.6 Service Deactivation (GGSN initiated) procedure	67
6.7 Service Deactivation (BM-SC initiated) procedure	69
6.8 De-registration (GGSN initiated) procedure.....	69
6.9 De-registration (BM-SC initiated) procedure.....	70
6.10 Closing connection	71
7. MBMS Evolution	72
7.1 MIMO Antennas.....	72
7.2 Dual receiver.....	74
7.3 MBMS on HSPDA (Adaptive Coding and Modulation)	75
7.4 Single Frequency Network (SFN)	76
7.5 Orthogonal Frequency-Division Multiple Access (OFDMA) Technology	77
8. IMS-MBMS integration.....	79
8.1 Introduction	79
8.2 Integration motivations.....	79
8.3 IMS architecture.....	80
8.4 MBMS-IMS integration analysis.....	81
8.5 The two candidate architecture approaches.....	82
8.5.1 IMS-MBMS integration proposal for 3GPP R7/R8 Network.....	82
8.5.2 IMS-MBMS integration proposal for an evolved/Converged Network.....	84
8.6 The Final Architecture.....	86
8.6.1 Key Service Enablers	87
8.6.2 Network Entities	88
8.7 IMS-MBMS Signalling.....	89
8.7.1 Service Activation	90
8.7.2 Session Start	91
8.7.3 Session Stop.....	93
8.7.4 Session Deactivation	93
9. Final Conclusions.....	95
References	97

List of Figures

Figure 1 - Mobile TV	14
Figure 2 - Signalling between GGSN and BM-SC	15
Figure 3 - Broadcast mode	18
Figure 4 - Multicast Mode	18
Figure 5 - Broadcast mode phases.....	19
Figure 6 - Multicast mode phases.....	19
Figure 7 – MBMS Reference Architecture Model	20
Figure 8 - MBMS Functional Structure	21
Figure 9 – MBMS Bearer Context.....	22
Figure 10 – Several MBMS UE Contexts	24
Figure 11 - Diameter Architecture.....	27
Figure 12 - A Diameter Proxy Agent.....	28
Figure 13 - A Diameter Redirect Agent.....	28
Figure 14 - A Diameter Translation Agent	28
Figure 15 - Diameter Message Format.....	29
Figure 16 - OpenBloX package architecture.....	31
Figure 17 – Activation of an MBMS multicast service.....	34
Figure 18 – MBMS Session Start procedure	35
Figure 19 – MBMS Session Stop Procedure	36
Figure 20 – MBMS Registration procedure	36
Figure 21 – MBMS De-Registration procedure.....	37
Figure 22 – MBMS De-Registration procedure (BM-SC initiated)	38
Figure 23 – MBMS Service Deactivation procedure.....	39
Figure 24 – MBMS Service Deactivation procedure (BM-SC initiated)	40
Figure 25 – Trace Session Activation Procedure	41
Figure 26 - Trace Session Deactivation Procedure	41
Figure 27 – MBMS UE Context Modification procedure	42
Figure 28 – Server's console	60
Figure 29 - Client's console	61
Figure 30 - Created MBMS Bearer Contexts.....	61
Figure 31 – GGSN sending a service activation command	62

Figure 32 - The BM-SC processing the service activation request.....	62
Figure 33 - Invalid requested service.....	62
Figure 34 - Service not found by the BM-SC	62
Figure 35 - The GGSN sends a create UE context authorization.....	63
Figure 36 - BM-SC creating the UE context	63
Figure 37 - UE Contexts stored in BM-SC.....	63
Figure 38 - Invalid APN in UE context creation request	64
Figure 39 - BM-SC informing GGSN that the APN is invalid	64
Figure 40 - IMSI not authorized	64
Figure 41 – BM-SC detecting that the IMSI is not yet authorized.....	64
Figure 42 – GGSN sending a registration request.....	65
Figure 43 - BM-SC adding the GGSN to the list of downstream nodes.....	65
Figure 44 - Displaying the MBMS Bearer Context.....	65
Figure 45 - BM-SC initiating Session Start procedure	66
Figure 46 - GGSN recognizing the Session Start procedure.....	66
Figure 47 - Request not sent to GGSN.....	66
Figure 48 - BM-SC initiating Session Stop procedure	67
Figure 49 - GGSN recognizing the Session Stop procedure	67
Figure 50 - Sending Deregistration request using a not recognized Session Id.....	67
Figure 51 - Invalid session Id.....	67
Figure 52 - Defining the new session Id and sending the request to the BM-SC	68
Figure 53 - BM-SC processing the leaving procedure.....	68
Figure 54 - GGSN sending the STR.....	68
Figure 55 - BM-SC releasing the UE context.....	69
Figure 56 - No UE contexts defined.....	69
Figure 57 - BM-SC initiating Service Deactivation procedure.....	69
Figure 58 - GGSN acknowledges the BM-SC request	69
Figure 59 - GGSN initiating De-registration procedure.....	70
Figure 60 - BM-SC removing the GGSN from the requested service	70
Figure 61 - Bearer context of the requested service.....	70
Figure 62 - BM-SC sending a service termination indication.....	71
Figure 63 - GGSN acknowledging the procedure.....	71
Figure 64 - BM-SC removing all information regarding the GGSN and the service	71

Figure 65 - Disconnecting.....	71
Figure 66 - Different configurations of multi-antenna.....	73
Figure 67 - Dual receiver process example	74
Figure 68 – Multicast/Broadcast SFN area.....	77
Figure 69 - Example allocation of user resources in OFDMA.....	78
Figure 70 - IMS Architecture.....	80
Figure 71 - R7/R8 IMS-MBMS Integration Architecture.....	82
Figure 72 - R7/R8 IMS-MBMS Network Entity Design	83
Figure 73 - Evolved IMS-MBMS Integration Architecture	84
Figure 74 - Evolved Integration Network Entity Design	85
Figure 75 – IMS-MBMS Network entity Design	87
Figure 76 - Service Activation Signalling	90
Figure 77 - Session Start Signalling	92
Figure 78 - Session Stop Signalling.....	93
Figure 79 - Session Deactivation Signalling	94

List of Tables

Table 1 – MBMS Bearer Context Parameters for GGSN and BM-SC.....	24
Table 2 - MBMS UE Context Parameters for GGSN and BM-SC	25
Table 3 - Diameter Base Protocol Messages	30

1. Introduction

File sharing, videoconferencing, e-learning, mobile TV (figure 1), radio, gaming – these are all applications that can benefit from broadcast/multicast. This is a well-established bandwidth-conserving technology that reduces the traffic in a network by allowing a host to transmit data to a transmission group instead of having to send all the packets to every single user, eliminating possible bottlenecks inside a network.



Figure 1 - Mobile TV

3GPP MBMS[2] (Multimedia/Broadcast and Multicast Service) is a broadcast/multicast technology in development within the 3GPP[23] (3rd Generation Partnership Project) that is specially designed for Universal Mobile Telecommunications System (UMTS) architecture. This technology is also available for Global System for Mobile Communications (GSM) networks.

The main difference between the broadcast and multicast mode is that a service using the former can be received without separate indication from users. Therefore, multicast users need a return channel to be able to subscribe to a particular desired service. And here is where the BM-SC comes into action.

The MBMS inserts into a GPRS architecture a new element as Broadcast Multicast – Serving Center (BM-SC). This network entity is responsible for transmitting MBMS services and storing user subscription information.

The BM-SC communicates with the rest of the mobile network via GGSN (Gateway GPRS Support Node) through two channels: Gi, the traffic channel that enables the two nodes to exchange MBMS service data, and Gmb, the control channel that handles the related signalling between them (figure 2).

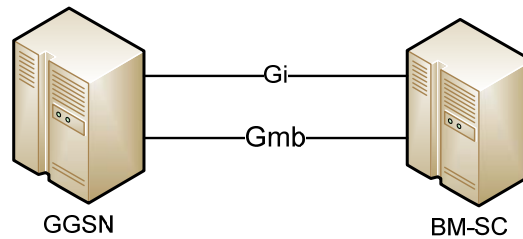


Figure 2 - Signalling between GGSN and BM-SC

The Gmb interface, the scope of this project, uses the Diameter protocol [3] to support the message exchange.

Thus, the main goal of this project is to implement the control interface Gmb, used by the GGSN and the BM-SC, using an existing Diameter base protocol software implementation.

In a second phase, MBMS evolution will be focused. There is always the need to MBMS to evolve due to an intense competition between technologies that enable the distribution of multimedia content through mobile network infrastructures.

Digital Video Broadcasting – Handheld (DVB-H) [25], one similar technology to MBMS, is the mobile version of the very successful DVB-T (Digital Video Broadcasting – Terrestrial) system for digital terrestrial television. Through a technique called time-slicing, bursts of data are sent periodically, allowing the user equipment to power off when it is inactive and therefore leading to power saving.

Another MBMS competitor, the Digital Multimedia Broadcasting (DMB) [28], is a digital radio transmission system that can operate via satellite (S-DMB) or terrestrial (T-DMB) transmission, also allowing the diffusion of multimedia content.

MediaFLO [26][27] (where FLO stands for Forward Link Only) is another similar technology where the multimedia content is sent on a frequency different from the ones used in current cellular networks, like DVB-H. In United States of America, where largely deployed, MediaFLO uses a previously allocated UHF TV frequency band and it is transmitted by a network of high-power broadcast transmitters.

However, MBMS has a crucial advantage over these rival technologies. Being an enhancement of the UMTS, MBMS can be easily integrated into the existing 2G/3G mobile network infrastructures. In contrary, technologies such as DVB-H, DBM and MediaFLO require significant investments in order to establish the coverage comparable to 2G/3G networks.

Despite this big benefit, it is important that MBMS design continues to evolve in order to maintain the competitiveness of Multicast and Broadcast systems. Therefore, a recent 3GPP study has considered several solutions based on the integration of MBMS with other technologies, such as Single Frequency Network (SFN), Multiple Input Multiple Output (MIMO), High-Speed Downlink Packet Access (HSDPA), Dual Receiver and Orthogonal Frequency-Division Multiplexing (OFDMA). Additionally, C-MOBILE project

proposed the integration of the IMS (IP Multimedia Subsystem) with the MBMS, aiming to obtain a global converged network that goes beyond the 3G.

In the next chapter, MBMS architecture will be referred to, with special focus on the GGSN and the BM-SC. In chapter 3, we will go through the Diameter protocol and how useful it is for this project implementation. This chapter will include an overview on the software structure used to implement the Gmb interface. After getting the basic notions on the Diameter, we will get a deeper view of this protocol into the Gmb interface context in chapter 4, along with all specific procedures and messages. Then, in chapter 5 the steps taken in the implementation of the interface will be described and in chapter 6 a demonstration of the final implementation will be carried out. Thereafter, in chapter 7, it will be described the most relevant approaches representing different technologies that, integrated with the concept of MBMS, characterize the state-of-the-art MBMS and its potential evolution. One of these approaches, the IMS-MBMS integration proposed by C-MOBILE, has a particular focus in chapter 8.

2. MBMS Architecture

2.1 Introduction

When many users in a UMTS want to access the same multimedia services at the same time, like weather forecast, news or sport services, UMTS radio resources bandwidth can run into shortage for a long period of time. Then, resource efficiency is a determinant factor and can be improved when many radio channels transfer the same information in parallel to different users and this within the same radio cell. Consequently, users might not subscribe to services when information transfer is delayed because of too many users.

The resource efficiency of multimedia services on mobile networks was the motivation behind the creation of MBMS (Multimedia/Broadcast and Multicast Service). MBMS is an IP datacast type of service, in which data is transmitted from a single source entity to multiple recipients, serving the content to large user groups simultaneously. Transmitting the same data to multiple recipients allows network resources to be shared.

The idea behind MBMS, standardized in various groups of 3GPP, is the seamless integration of broadcast/multicast transmission capabilities into 3G service and network infrastructure, like GSM and UMTS cellular networks.

2.2 MBMS Bearer Service and User Service

MBMS is split into the MBMS Bearer Service and the MBMS User Service.

MBMS User Service consists in applications which offer, for example, multimedia content to the users and can be classified into three groups as follows:

- Streaming services, like audio and video streaming
- File download services
- Carousel service, a combination of streaming and file download

MBMS User Services are based on the transport services provided by the MBMS Bearer Service. The latter is designated to deliver IP multicast datagrams to multiple receivers, easing provision of existing applications in the mobile environment and simplifying the creation of new applications. MBMS BS includes a broadcast mode and multicast mode.

2.3 Broadcast and Multicast Mode

In MBMS two modes of operation are defined: broadcast and multicast.

Generally, broadcast is a unidirectional point-to-multipoint service in which multimedia data is transmitted from a single source to multiple terminals. Data is transmitted over a common radio channel and is defined by the network. It is delivered to a specified area without knowing the receivers and whether there is any receiver at all in this area (Figure 3).

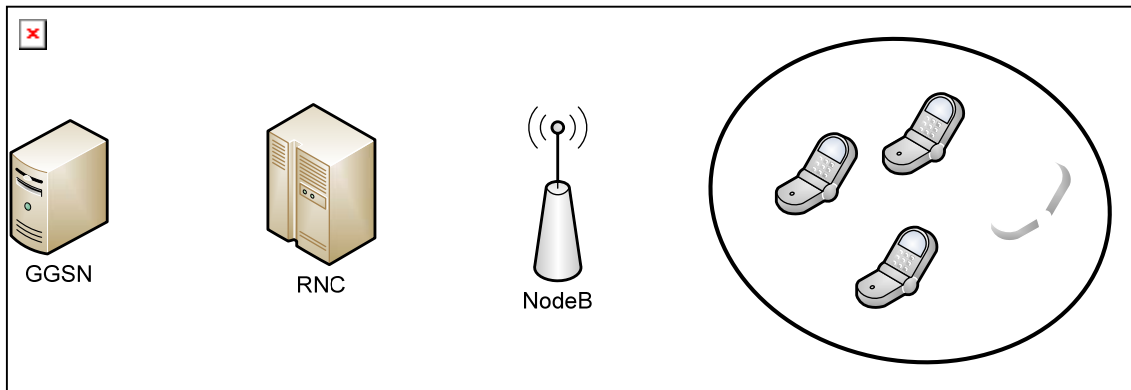


Figure 3 - Broadcast mode

The broadcast mode is expected to be a service without charging and there are no specific activation requirements for this mode.

On the other hand, multicast mode can be defined as a unidirectional point-to-multipoint transmission of multimedia data from a single source point to a multicast group in a multicast service area. This service mode generally requires subscription to the multicast subscription group and then the user, after monitoring service announcements regarding service availability, joins the corresponding multicast group (Figure 4).

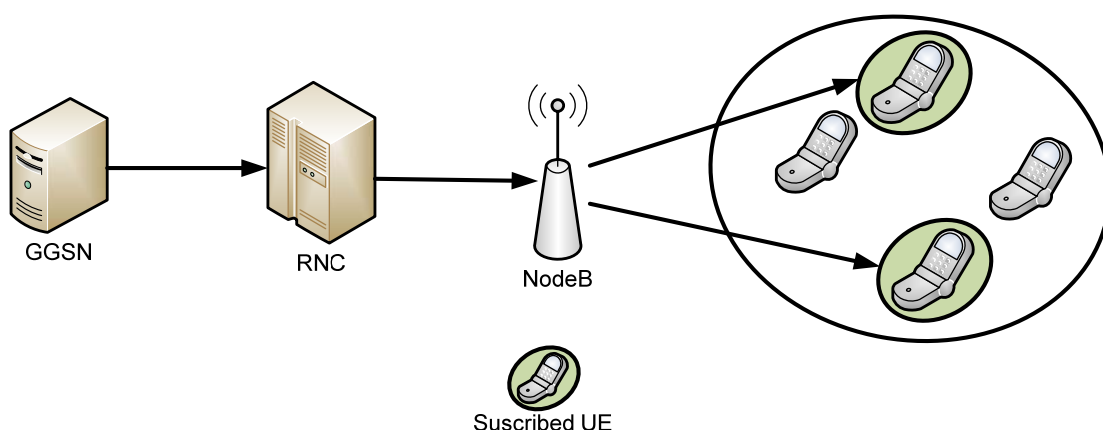


Figure 4 - Multicast Mode

A multicast service might consist of a single on-going session or several intermittent multicast sessions over an extended period of time.

This mode is suitable for pay-per-view or pay-per-program services such as television or video on demand and it is expected that charging data for the end user will be generated. Reception of both MBMS broadcast and multicast modes are enabled by certain procedures, as shown in the two figures below:

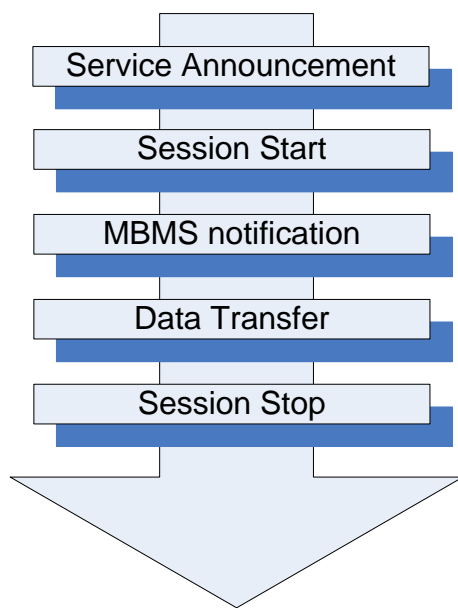


Figure 5 - Broadcast mode phases

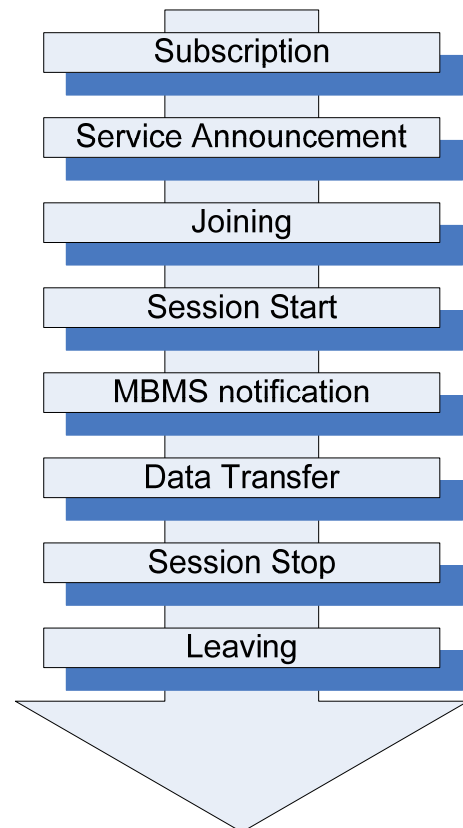


Figure 6 - Multicast mode phases

The main difference between these two modes, looking at the phases of each one, is that in case a user is interested in a particular multicast service, he needs to be subscribed to that service (Subscription) and later join it (Joining) to be able to receive the data from the service. Later, the user could leave anytime he wants (Leaving). Instead, in a broadcast service, all users get the service without the need for being subscribed.

2.4 MBMS Architecture Model

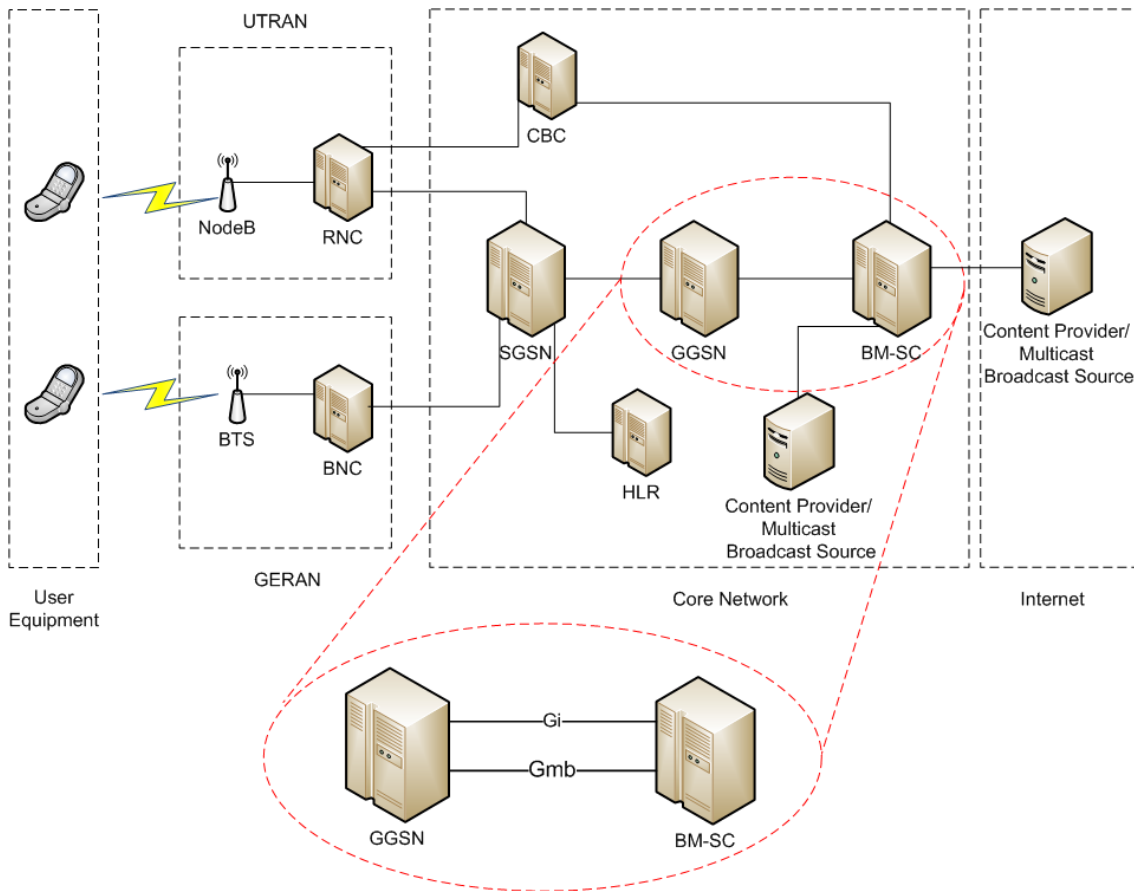


Figure 7 – MBMS Reference Architecture Model

Regarding the basic UMTS core network architecture, a completely new functional entity is added: the BM-SC serving as central controlling unit (Figure 7). It is responsible for providing membership functions, security functions, proxy and transport functions and control session establishment and data delivery over the two added reference points Gmb and Gi.

The Gateway GPRS Support Node (GGSN), the Serving GPRS Support Node (SGSN), the Radio Network Controller (RNC), the Node-B and the User Equipment (UE) are the existing UMTS network elements that are modified so as to handle the establishment of multicast distribution trees and the transmission of MBMS data over these trees.

The Gi reference point between BM-SC and GGSN enables the BM-SC to exchange MBMS service data with the GGSN, while the Gmb enables the BM-SC to exchange MBMS service control information with the GGSN, being the Gmb the scope of this project.

2.5 GGSN

The GGSN (Gateway GPRS Support Node) is a network node that provides an interface between mobile networks and the Internet or corporate intranets.

In a MBMS point of view, GGSN serves as an entry point for IP multicast traffic as MBMS data. GGSN is able to request the establishment of a bearer plane for a broadcast or multicast transmissions upon BM-SC. Then, those transmissions are carried out towards the SGSN's that requested to receive that specific MBMS Bearer Service.

2.6 BM-SC

The BM-SC is the MBMS data source. It includes functions for MBMS user service provisioning and delivery. BM-SC may serve as an entry point for MBMS content providers, used to authorize and initiate MBMS Bearer Services within the PLMN. In addition, it can be used to schedule and deliver MBMS transmissions.

BM-SC is a functional entity that consists of five sub-functions, represented in Figure 8:

- Membership function
- Session and Transmission function
- Proxy and Transport function
- Service Announcement function
- Security function

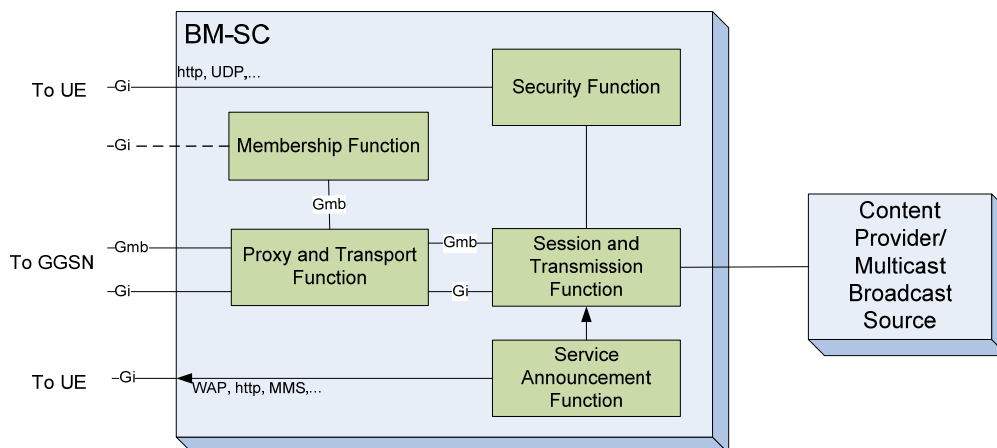


Figure 8 - MBMS Functional Structure

The BM-SC Membership Function role is to provide authorization for each UE that requests the activation of a MBMS service. It stores all information regarding subscription data and charging for MBMS service users.

The schedule of MBMS session transmissions is done by the BM-SC Session and Transmission Function. This sub-function also informs the GGSN about the transport associated parameters (like the quality of service and the service area) and authenticates/authorizes the content of external sources.

BM-SC Proxy and Transport Function is basically a Proxy Agent for signalling over Gmb reference point between GGSN and other BM-SC sub-functions.

Service announcements and media descriptions (like the type of video or audio encodings) related to multicast and broadcast MBMS user services is the responsibility of the BM-SC Service Announcement function.

And finally, MBMS Security Function is used for security issues regarding integrity and confidentiality protections of MBMS data.

2.7 MBMS Attributes and Parameters

Both GGSN and BM-SC, like the other network nodes, must maintain two types of state: packet forwarding state, allowing them to determine which other node should receive a packet, and user state, allowing the network to charge the participating receivers.

Each such entity, therefore, maintains a MBMS Bearer Context for each multicast group present in its area and a MBMS UE Context for each UE served by the node that is currently a member of such a group.

2.7.1 MBMS Bearer Context

The MBMS Bearer Context contains all information describing a particular MBMS bearer service and it's created in each node involved in the delivery of the MBMS data.

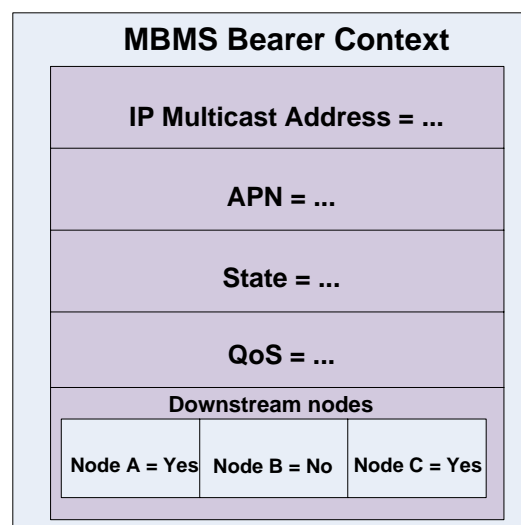


Figure 9 – MBMS Bearer Context

For example, in Figure 9, representing a MBMS Bearer Context, only the nodes A and C can receive data from that particular service identified by an IP multicast address.

The content of the MBMS Bearer Context is described in Table 1.

Parameter	Description	GGSN	BM-SC
Multicast/broadcast mode	MBMS bearer service in broadcast or multicast mode	X	X
IP multicast address(multicast mode only)	IP multicast address identifying the MBMS bearer described by this MBMS Bearer Context.	X	X
APN (multicast mode only)	Access Point Name on which this IP multicast address is defined.	X	X
TMGI	Temporary Mobile Group Identity allocated to the MBMS bearer service.	X	X
State	State of bearer plane resources ('standby' or 'active')	X	X
Required MBMS Bearer Capabilities (multicast mode only)	Minimum bearer capabilities the UE needs to support	X	X
QoS	Quality of Service required for the MBMS bearer service.	X	X
MBMS Service Area	Area over which the MBMS bearer service has to be distributed.	X	X
List of downstream nodes	List of downstream nodes that have requested the MBMS bearer service and to which notifications and MBMS data have to be forwarded.	X	X
Number of UEs (multicast mode only)	Number of UEs hosted by the node that have joined the multicast MBMS bearer service.	X	

Table 1 – MBMS Bearer Context Parameters for GGSN and BM-SC

2.7.2 MBMS UE Context

The MBMS UE Context contains UE-specific information related to a particular MBMS bearer service that the UE has joined. It is created in BM-SC and in the GGSN at the time of the join.

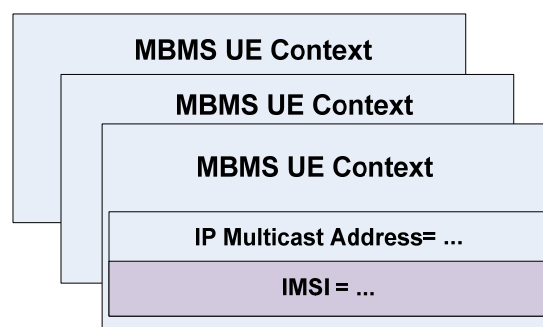


Figure 10 – Several MBMS UE Contexts

Figure 10 shows several MBMS UE Contexts related to a particular MBMS bearer service.

The content of the MBMS UE Context is described in Table 2.

Parameter	Description	GGSN	BM-SC
IP multicast address	IP multicast address identifying an MBMS bearer that the UE has joined	X	X
APN	Access Point Name on which this IP multicast address is defined	X	X
SGSN address	The IP address of SGSN	X	
IMSI	IMSI identifying the user	X	X
MBMS_NSAPI	Network layer Service Access Point Identifier which identifies an MBMS UE Context	X	
Additional MBMS Trace Info	Identifies additional information required to perform trace	(1)	(1)
Trace Reference	Identifies a record or a collection of records for a particular trace	(1)	
Trace Type	Indicates the type of trace	(1)	
Trigger Id	Identifies the entity that initiated the trace	(1)	
OMC Identity	Identifies the OMC that shall receive the trace record(s)	(1)	

(1) Trace parameters are only stored if trace is activated

Table 2 - MBMS UE Context Parameters for GGSN and BM-SC

2.8 Gmb Interface

Signalling between BM-SC and GGSN is exchanged via Gmb interface, which includes the user specific and bearer service messages.

In a bearer service messages point of view, the GGSN can register or de-register at BM-SC. For instance, the BM-SC can de-register the GGSN from itself and is able to indicate a session start or stop to the GGSN, including session attributes like QoS and MBMS Service Area.

As for the user specific signalling messages, if there is a user that wants to join a particular service, the BM-SC proceeds to the authorization. If the authorization is granted, the GGSN reports to the BM-SC the success of the user activation. In addition, the GGSN informs the BM-SC about the fact that a specific MBMS multicast service user is released or deactivated.

The standardisation organization 3GPP has adopted the Diameter Protocol [3] as the primary signalling protocol for AAA for many reasons. It is specially designed for network access, supports IP mobility and it is also adequate for future extensions to new technologies. Therefore, the control information between the BM-SC and the GGSN is done using Diameter, described in the next chapter.

3. The Diameter Protocol

3.1 Description

The Diameter protocol [3] basic concept is to provide an Authentication, Authorization and Accounting (AAA) framework for applications such as network access or IP mobility. It was derived from the RADIUS (Remote Authentication Dial-In User Service) Protocol, with a lot of improvements.

3.2 History

The RADIUS protocol has been widely and successfully deployed to provide authentication, authorization, and accounting (AAA) services for dial-up PPP/IP and other similar technologies.

However, due to new applications and technologies like mobile IP's and wireless networks emerging in our technology world, the requirements for authentication and authorization have greatly increased. So, the natural imperfections of the RADIUS protocol have limited its ability to adapt to the growing capabilities of routers and network access servers, and the ever-expanding set of desired AAA services. What is needed is a new protocol that is capable of fulfilling new access control features while keeping the flexibility for further extension. And this is the role of the Diameter Protocol.

3.3 Advantages of Diameter over RADIUS

The name Diameter is a pun on the RADIUS Protocol [24], which is the predecessor (the diameter is twice the radius) and came to solve the known RADIUS problems today.

Diameter supports capability negotiations and better roaming support, ideal for mobile communications. This protocol can also run over a much more reliable transport, like TCP and STCP. Lost packets are retransmitted and the data flow adapts to network congestion. Transport failures can now be detected and retransmitted at the appropriated places, resulting in a better proxying. There is also a better session control due to the independence of accounting over session management. This means that accounting information can be routed to a different server than authentication/authorization messages. In addition, Diameter supports IPSec and TLS, assuring integrity and confidentiality of the messages, resulting in a better security. Finally, the interoperability of Diameter over Radius: the latter only supports vendor specific attributes. Vendors had to create private command codes ensuing in interoperability problems. With Diameter supporting vendor-specific commands, this problem is solved.

3.4 The Diameter Architecture

Diameter is defined by a set of messages that represent the core of the base protocol and a set of applications that require AAA functions. This architecture (Figure 11) allows the extension to new access technologies, each one relying on the services of the Diameter Base protocol to support.

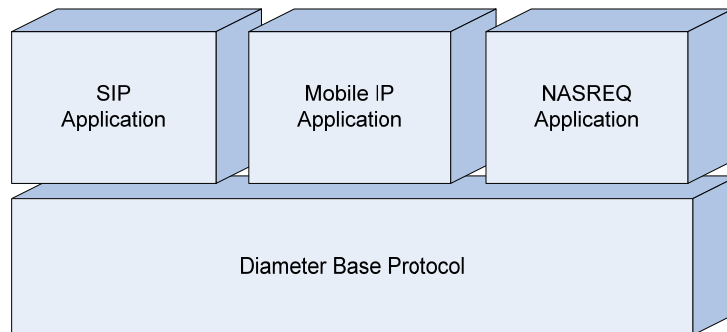


Figure 11 - Diameter Architecture

3.5 Diameter Nodes

Diameter is a Peer-to-Peer architecture and every host that takes part on this architecture is a Diameter node. Each host can be a server, a client or an agent:

- **Client** – device at the edge of the network that performs access control. It generates Diameter messages to request authentication, authorization, and accounting services for the user.
- **Server** – node that performs authentication and/or authorization of the user for a particular realm.
- **Relay Agent** – Diameter agents that receive request messages and route them to other nodes based on information found in those messages. This routed decision is based on a list of supported realms and known peers. Relay agents can modify routing information on the message but can not modify other portion of the message.
- **Proxy Agent** – similar to Relay agents, but with the main difference that they can modify messages to implement policy enforcements (Figure 12).
- **Redirect Agent** – node that acts as a centralized configuration repository for other Diameter nodes. It returns a special type of answer message to the requesting peer that contains routing information that allows the peer to send the request again directly to the correct destination (Figure 13).
- **Translation Agent** – responsible for converting a message from one AAA protocol to another, such as RADIUS and Diameter (Figure 14).

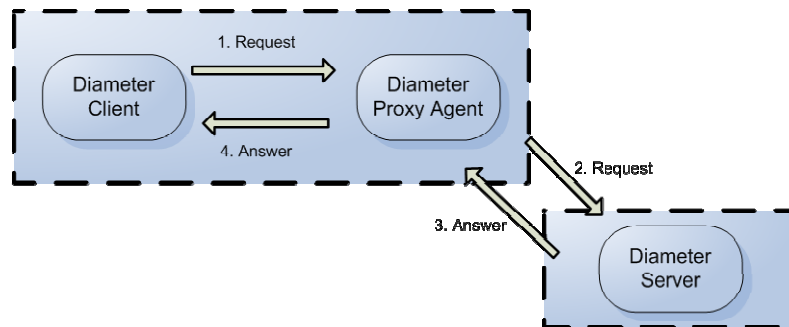


Figure 12 - A Diameter Proxy Agent

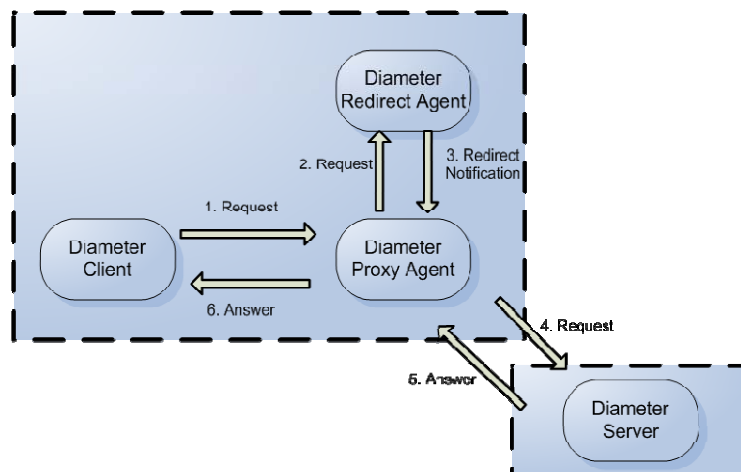


Figure 13 - A Diameter Redirect Agent

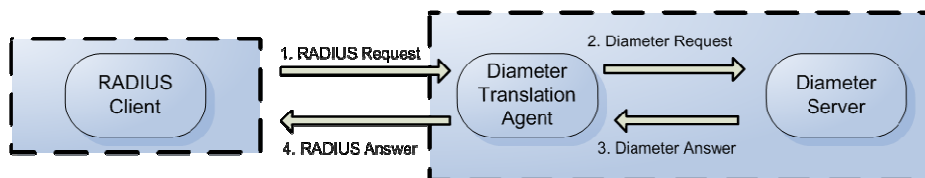


Figure 14 - A Diameter Translation Agent

3.6 Diameter Messages

A Diameter message is the base unit to send a command or a notification to other Diameter node. Due to the synchronous message exchange style, when a node sends a request, this node prepares to receive an answer.

Diameter has defined several types of messages, like Capability-Exchange-Request/Answer and Accounting-Request/Answer. The type of the message is defined by a command code and the actual data, carried by a set of AVP's (Attribute Value

Pairs), could be details of AAA, routing, security and capability information. Each AVP is associated with an AVP Data Format (for example OctetString, UTF8String, Integer32) that must be followed.

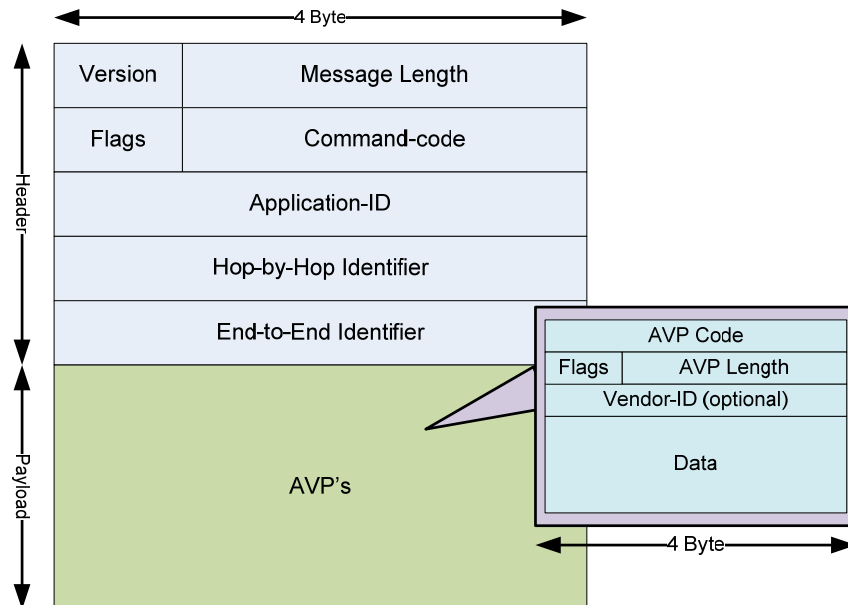


Figure 15 - Diameter Message Format

A Diameter message header contains the Version field, containing the Diameter version, three octets indicating the message length, an octet containing flags and a Command-code communicating the command associated with the message. The next three 32-bit fields are Vendor-ID, used to identify which application the message is applicable for, Hop-by-Hop identifier that aids in matching requests and replies, and finally the End-to-End identifier used to detect duplicate request messages.

Regarding the AVP format, the AVP code field combined with the Vendor-Id field identifies the attribute uniquely. There is also the Flags field, the AVP Length field and the AVP Data field, this one corresponding to zero or more octets containing information specific to the Attribute.

The table below shows all Diameter base protocol defined messages:

Message Name	Abbrev	Code
Abort-Session-Request	ASR	274
Abort-Session-Answer	ASA	274
Accounting-Request	ACR	271
Accounting-Answer	ACA	271
Capabilities-Exchanging-Request	CER	257
Capabilities-Exchanging-Answer	CEA	257
Device-Watchdog-Request	DWR	280
Device-Watchdog-Answer	DWA	280
Disconnect-Peer-Request	DPR	282
Disconnect-Peer-Answer	DPA	282
Re-Auth-Request	RAR	258

Re-Auth-Answer	RAA	258
Session-Termination-Request	STR	275
Session-Termination-Answer	STA	275

Table 3 - Diameter Base Protocol Messages

3.7 The OpenBloX™ Implementation



This section focuses on the software implementation used to reach the goal of this project: to develop the signalling between the GGSN and the BM-SC over the Gmb interface. Without going into many technical details, a short description of this Diameter base protocol implementation will be made.

3.7.1 Overview

The OpenBloX™ framework [4], by Traffix Systems, is an Open Source set of directories and files, implementing on a whole or part of the 3GPP and 3GPP2 Diameter specifications, which provides a convenient way of using the Diameter stack in a Java environment. The package contains, at minimum, the Diameter base protocol as described by IETF RFC 3588 and any extensions provided to support upper layers as specified by the 3GPP specifications, such as Rx, Gx, Ro, Cx, Sh and Gmb itself. Note that having this upper layer support does not mean that these specifications are implemented, but rather the fact that its design contains utilities that are set to support further 3GPP defined extensions.

The OpenBloX™ is an object oriented package, developed in Java and is built upon a strong structure. It is system and operating system agnostic, has a high efficiency/scalability and supports both TCP and SCTP.

3.7.2 Reasons for its use on this work

When having to develop an implementation interface like Gmb, a strong software structure is strictly necessary to provide a reliable way of extending further applications to a higher level, and that is the case of the OpenBloX package.

From all the available Diameter protocol Open Source implementations, OpenBloX seemed to be the most complete and the easiest one to extend. Before choosing this set of software as the structure of development for this project, two other distinct Diameter implementations were studied: OpenDiameter (C++) and JavaDiameterPeer (Java).

The main reason why OpenBloX stood out from these different applications is the fact that it better supports the development of the Gmb specification.

3.7.3 OpenBloX™ Architecture

The OpenBloX™ package consists on three main parts: the Diameter Base protocol (complying with RFC 3588 and RFC 3539), the support for Diameter 3GPP/3GPP2 extensions and an API based on JDiameter (*de facto* industry standard API, the leading solution for upgraded usability and maintenance).

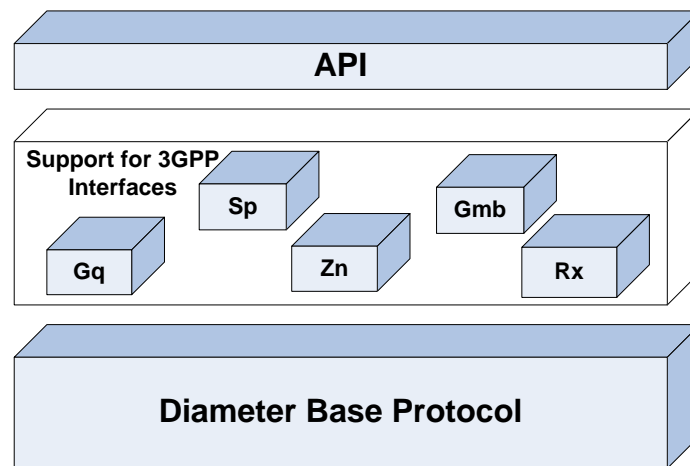


Figure 16 - OpenBloX package architecture

4. Diameter on Gmb

Signalling between GGSN and BM-SC is exchanged at Gmb reference point. The interface is used to:

- Request and answer for authorization/deactivation of a user for a multicast MBMS service;
- Register/de-register the GGSN for receiving a multicast MBMS service;
- Send and receive indication of a session start/stop messages;
- Indicate that a user is being removed from a particular multicast MBMS bearer service.

4.1 MBMS Procedures on Gmb

4.1.1 MBMS User Authorization

Upon reception of a Join message for an IP multicast address identifying a MBMS service, the GGSN requests authorization of the user for this MBMS bearer service.

4.1.2 MBMS service registration/de-registration

After the authorization of the first user on a particular GGSN, the MBMS service registration of the GGSN at the BM-SC is performed for that particular MBMS service. Afterwards, when the last user leaves a GGSN for a particular service, the MBMS service de-registration takes place. In case a specific service is terminated, the BM-SC initiates the service de-registration procedure.

4.1.3 MBMS session start/stop

To announce the arrival of data for a MBMS service, BM-SC initiates the session start procedure and discloses its attributes, like QoS and service area, to all GGSN present in its downstream nodes list. The session stop procedure is initiated by the BM-SC to indicate the end of that streaming data.

4.1.4 MBMS User deactivation

The MBMS user deactivation is a procedure that removes all information in the MBMS UE Context regarding a particular user. This happens when that user leaves a MBMS service and can be initiated by the GGSN or the BM-SC.

4.2 Gmb Message Flows

This section allows a better understanding of how the procedures take place on the Gmb interface. These message flows, presented next in form of flow diagrams, represent a set of actions that occur between the GGSN and the BM-SC, along with the type of Gmb messages used in these actions (these type of messages are described in section 4.3), showing a set of procedures that define the Gmb signalling.

4.2.1 Service Authorization Procedure

This procedure registers the user that requests to join a specific MBMS service, so it can receive this MBMS data content over the network.

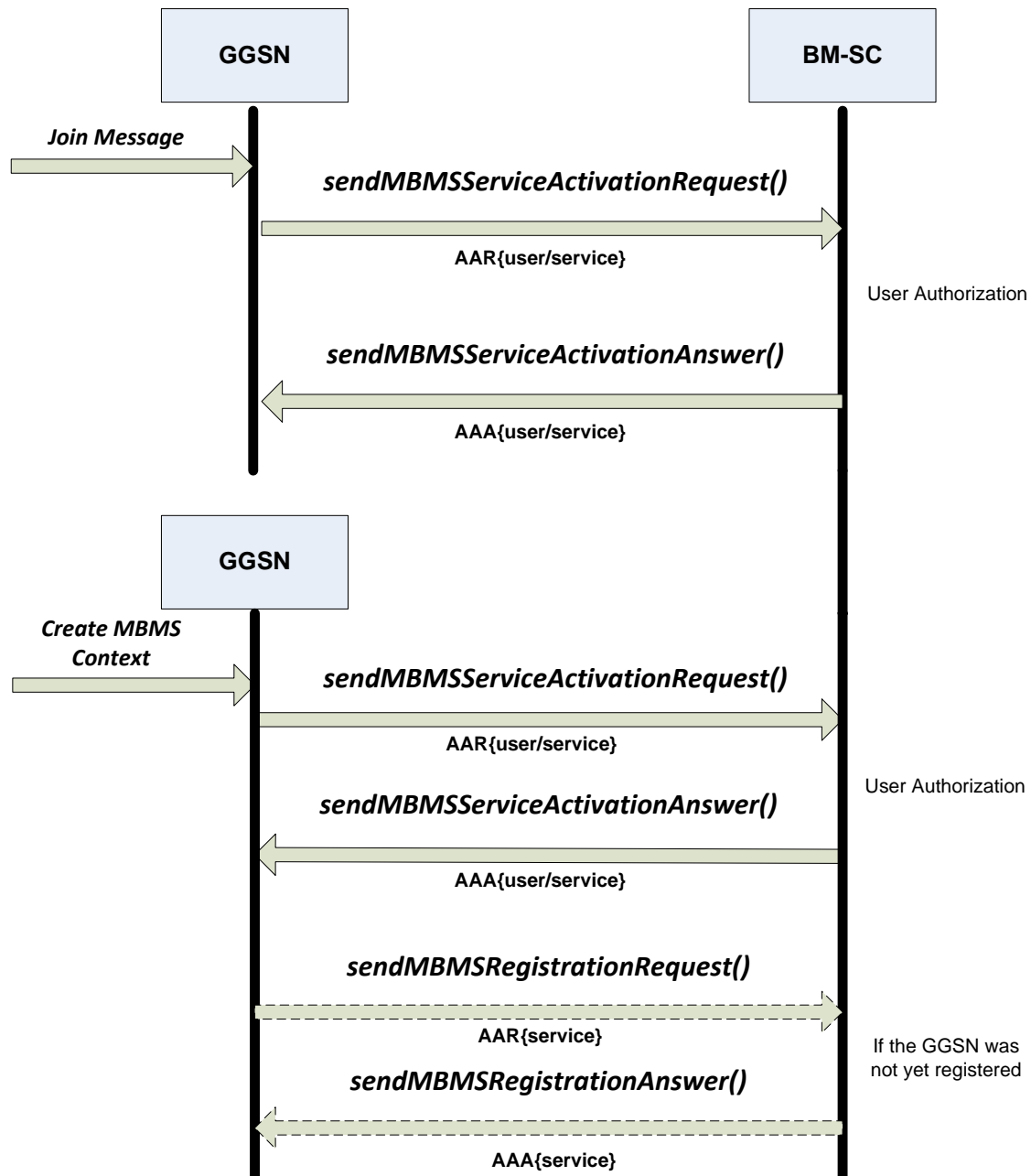


Figure 17 – Activation of an MBMS multicast service

When a UE wants to receive data from a particular MBMS service, it sends a join message to the GGSN, indicating the requested service. Then, the GGSN seeks authorization from the BM-SC for the activating UE to receive data from the service. The BM-SC provides the decision together with the APN to be used in the creation of the MBMS UE Context. If the request is denied, there is no additional message exchange.

Then, the GGSN (that can be different from the one that received the join message) sends a request seeking authorization for the creation of a UE Context for that user. The decision is provided by the BM-SC and, if authorized, it creates the UE Context information.

Conditionally, if the requesting GGSN is not yet registered in the BM-SC for the service, the registration procedure is initiated. See Registration Procedure message flow (4.5.4).

4.2.2. Session Start Procedure

The BM-SC initiates this procedure when it is ready to send data, informing all GGSN present in its downstream nodes list the imminent start of a transmission. The session attributes are also provided.

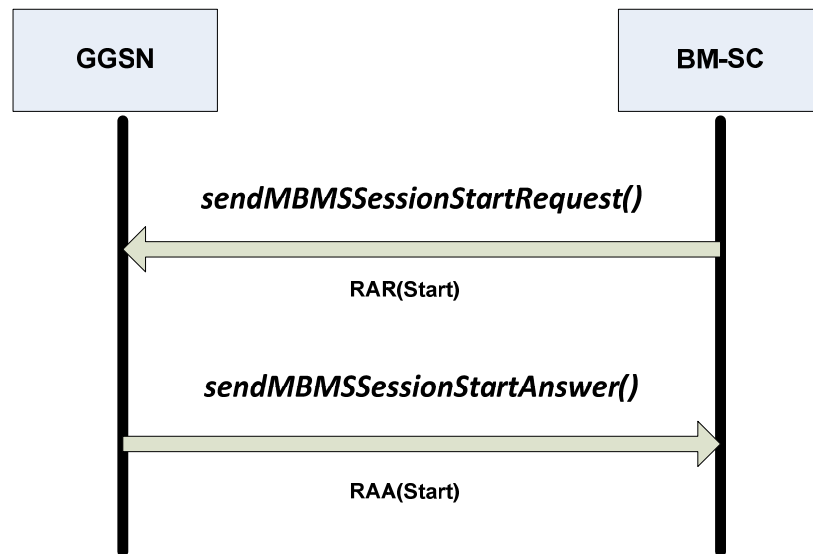


Figure 18 – MBMS Session Start procedure

The BM-SC indicates an impending start of the transmission sending a session start request to all GGSN that are registered in that particular service. That service changes its context state to 'Active', i.e., data is being sent over the network. Then, the GGSN acknowledges the imminent start for that service sending an answer message to the BM-SC.

4.2.3 Session Stop Procedure

The Session Stop procedure occurs when the BM-SC considers the active service to be terminated.

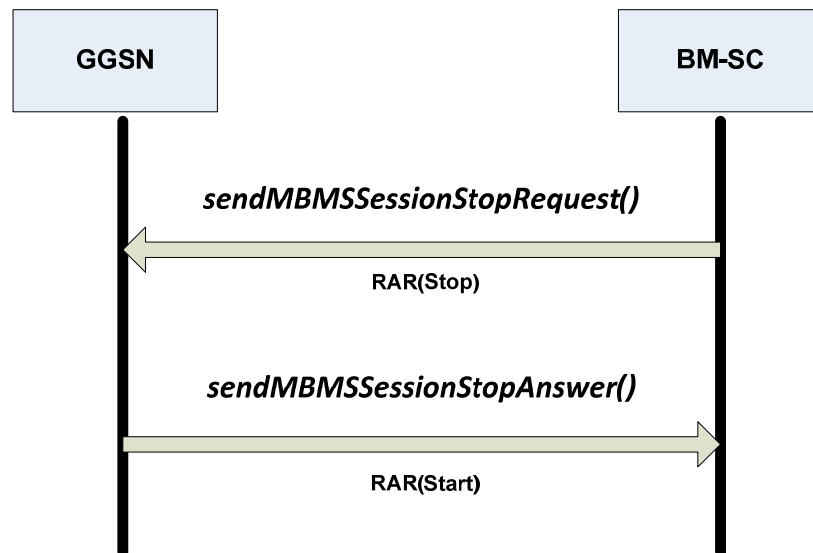


Figure 19 – MBMS Session Stop Procedure

The BM-SC sends a session stop request message to all GGSN's listed in its list of downstream nodes of the affected MBMS Bearer Context to indicate that the MBMS session is terminated. The service changes to 'Stand-By' state. Afterwards, the GGSN sends an answer message to the BM-SC acknowledging the termination of data transmission.

4.2.4 Registration Procedure

This procedure takes place when the GGSN indicates the BM-SC that it would like to receive data from a particular multicast MBMS bearer service, in order to distribute to interested users.

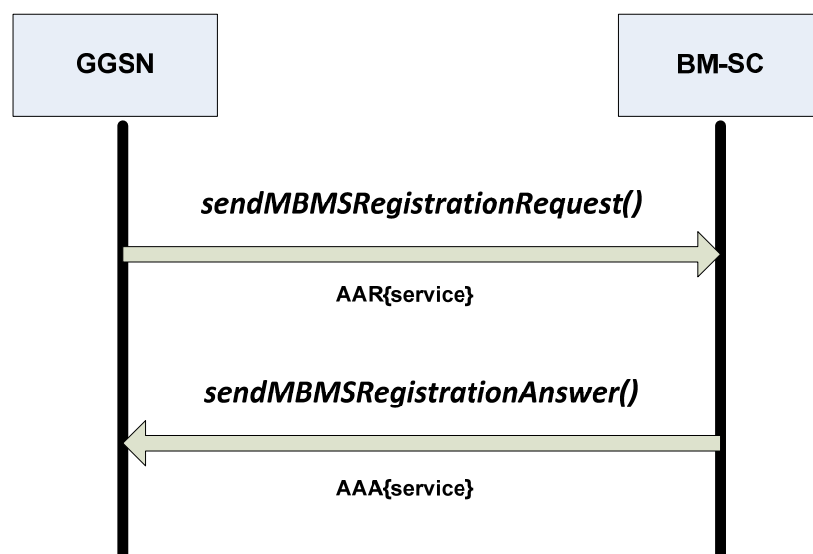


Figure 20 – MBMS Registration procedure

When a user joins a particular MBMS service and the affected GGSN is not yet registered in the BM-SC, the GGSN sends a message requesting that registration, indicating the service IP and the APN. Upon reception of the request, the BM-SC adds the identifier of the GGSN to its list of downstream nodes and responds with an answer message containing the service required capabilities, among other related information.

Additionally, if the service is being transmitted at the moment of the registration, Session Start procedure is initiated (4.2.2).

4.2.5 De-registration Procedure (GGSN Initiated)

The de-registration procedure occurs when the GGSN informs the BM-SC that it does not need to receive data for a particular multicast MBMS bearer service anymore.

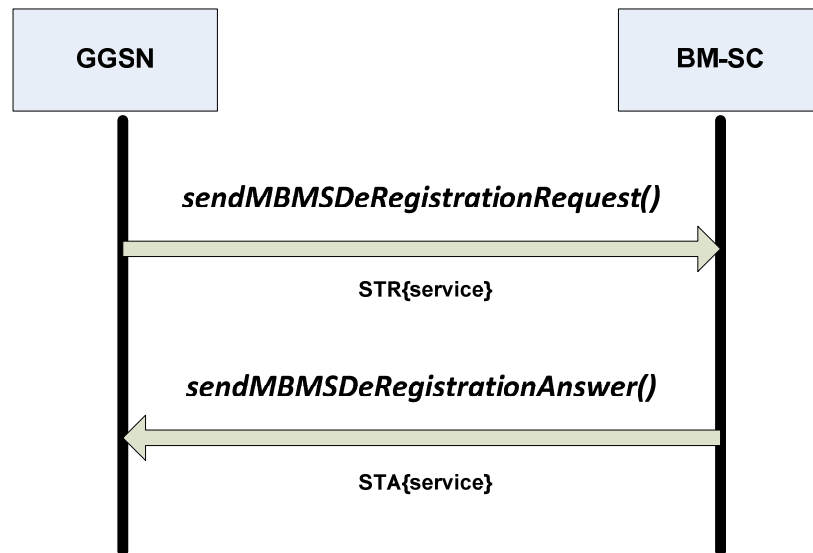


Figure 21 – MBMS De-Registration procedure

When the GGSN has no users interested in a particular MBMS service, it sends a message requesting that it would like to be removed from the BM-SC downstream nodes list. After the deletion, the BM-SC confirms the procedure with an answer message.

4.2.6 De-registration Procedure (BM-SC initiated)

There is also a situation in which the BM-SC is the one that takes the initiative to de-register the GGSN when a particular MBMS service is terminated. This procedure results in releasing all MBMS Bearer Contexts.



Figure 22 – MBMS De-Registration procedure (BM-SC initiated)

The BM-SC sends a message to all GGSN's contained in its list of downstream nodes of the corresponding MBMS Bearer Context to indicate that a specific MBMS service is terminated. After the corresponding answer, the BM-SC releases all UE Contexts and removes the GGSN from its list.

4.2.7 Service Deactivation Procedure

The multicast service deactivation is a procedure that will terminate the user registration to a particular MBMS multicast service. The multicast service deactivation can be initiated by the GGSN, when indicated so by the UE, or by the BM-SC, for service specific reasons.

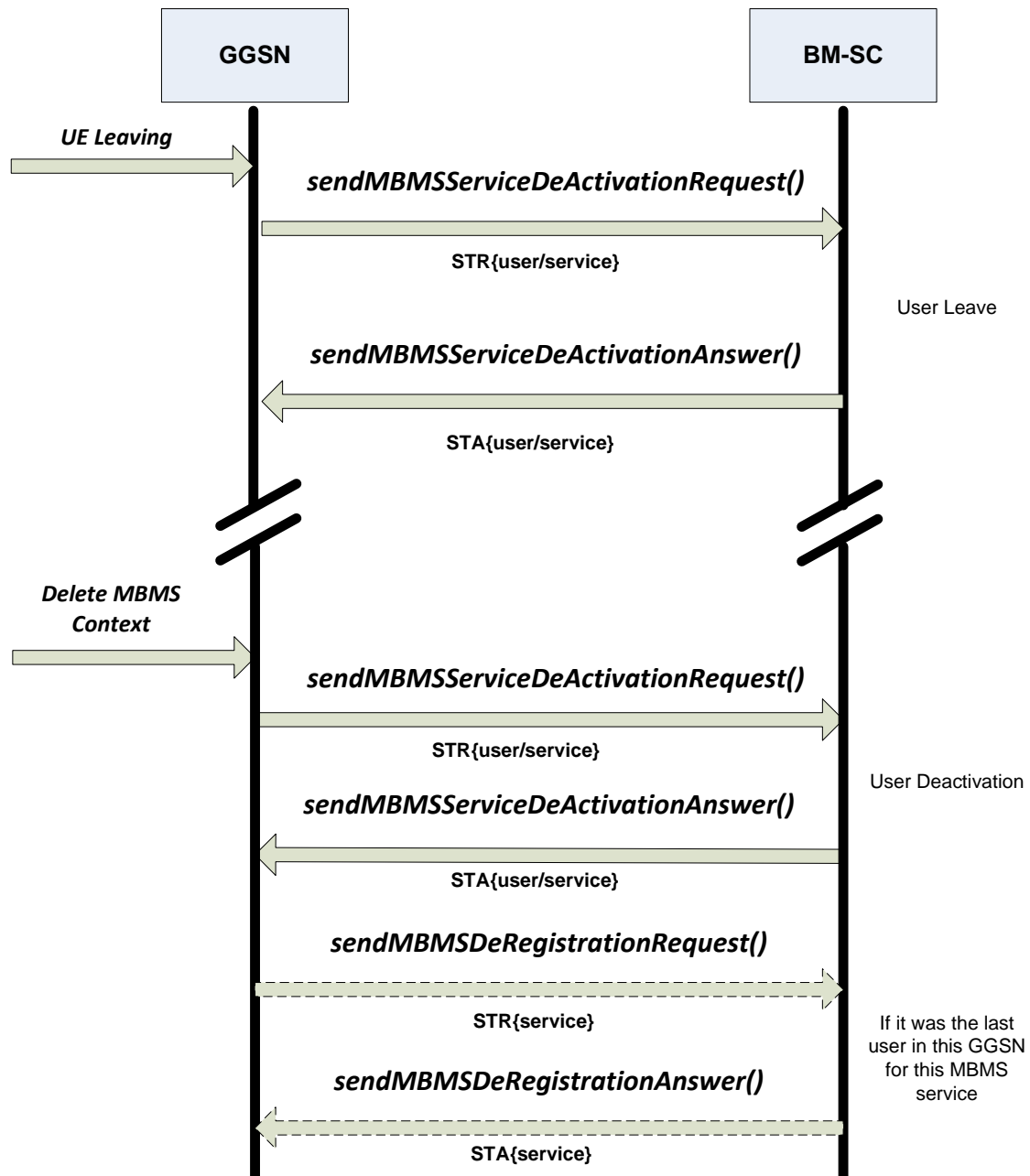


Figure 23 – MBMS Service Deactivation procedure

If a user sends a message to leave a particular multicast service, the GGSN sends a request message to the BM-SC indicating that the user is leaving the MBMS service. Upon reception, the BM-SC verifies the service and sends an answer to the GGSN that originated the leave indication.

After that, the GGSN (that could be different from the one that received the leave indication from the user) sends a message to the BM-SC, confirming the successful deactivation. Then, the BM-SC deletes the MBMS UE Context and sends a confirmation to the GGSN in an answer message.

If the GGSN does not have any more users interested in this MBMS bearer service, it initiates a De-Registration procedure (4.2.5).

4.2.7.1 Service Deactivation Procedure (BM-SC initiated)

For service specific reasons, the BM-SC can be the entity to initiate the Service Deactivation procedure.

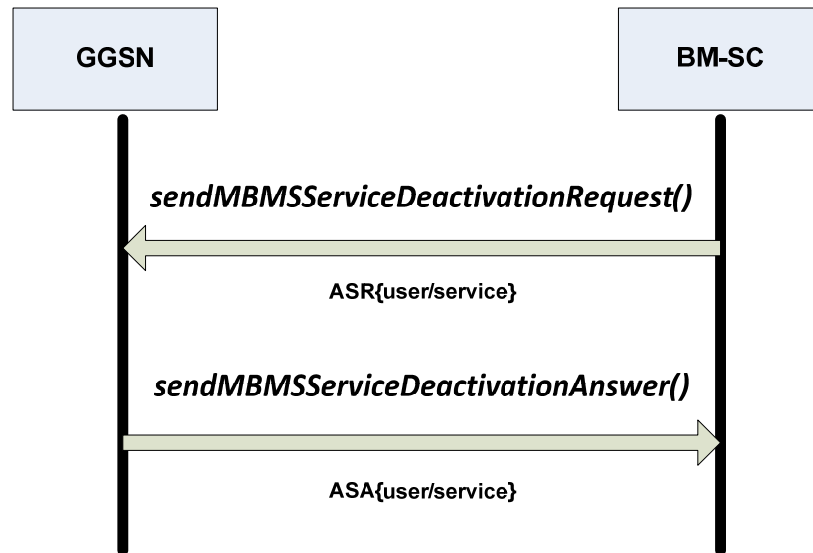


Figure 24 – MBMS Service Deactivation procedure (BM-SC initiated)

The BM-SC sends a request message to the GGSN, indicating that the UE shall be removed from the multicast service. The GGSN then sends a message back confirming the request.

From this point forward, the same occurs as the GGSN initiated Service Activation: the GGSN (which could be different from the one that received the leave indication from the user) sends a message to the BM-SC confirming the successful deactivation. Then, the BM-SC deletes the MBMS UE Context and sends a confirmation to the GGSN in an answer message.

If the GGSN does not have any more users interested in this MBMS bearer service, it initiates a De-Registration procedure (4.2.5).

4.2.8 Trace Session Activation Procedure

The Trace Session Activation procedure occurs when the GGSN indicates to the BM-SC that a Trace Session needs to be activated.

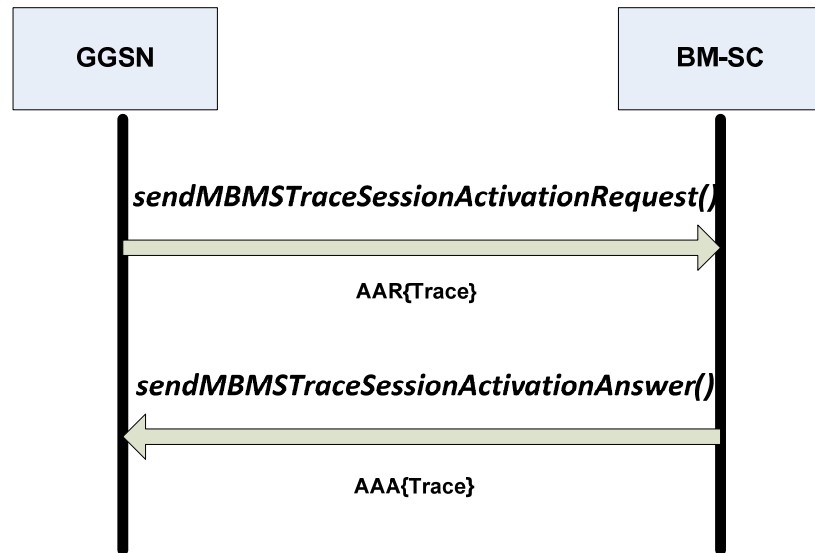


Figure 25 – Trace Session Activation Procedure

When the GGSN has received a Trace Activation message from the SGSN, in a Create MBMS Context Request/Update MBMS Context Request, which requires the activation of a Trace Session in the BM-SC, the GGSN sends a request message to activate a trace session in the BM-SC. Upon reception, the BM-SC responds with an answer message.

4.2.9 Trace Session Deactivation

The Trace Session Deactivation procedure occurs when the GGSN indicates to the BM-SC that a Trace Session, previously activated, needs to be deactivated.

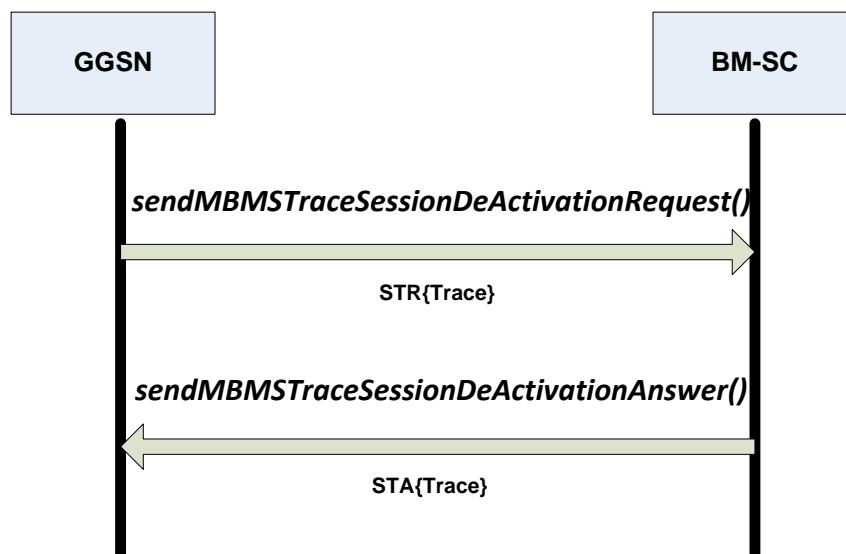


Figure 26 - Trace Session Deactivation Procedure

When the GGSN has received a Trace Deactivation message from the SGSN, in a Create MBMS Context Request/Update MBMS Context Request, which requires the deactivation of a Trace Session in the BM-SC, the GGSN sends a request message to deactivate a trace session in the BM-SC. Upon reception, the BM-SC responds with an answer message.

4.2.10 UE Context Modification

During the multicast MBMS bearer service activation, the MBMS UE Context is stored in the BM-SC. Later, the MBMS UE Context shall be updated when the UE enters a new routing area. This procedure enables the BM-SC to update its MBMS UE context accordingly.

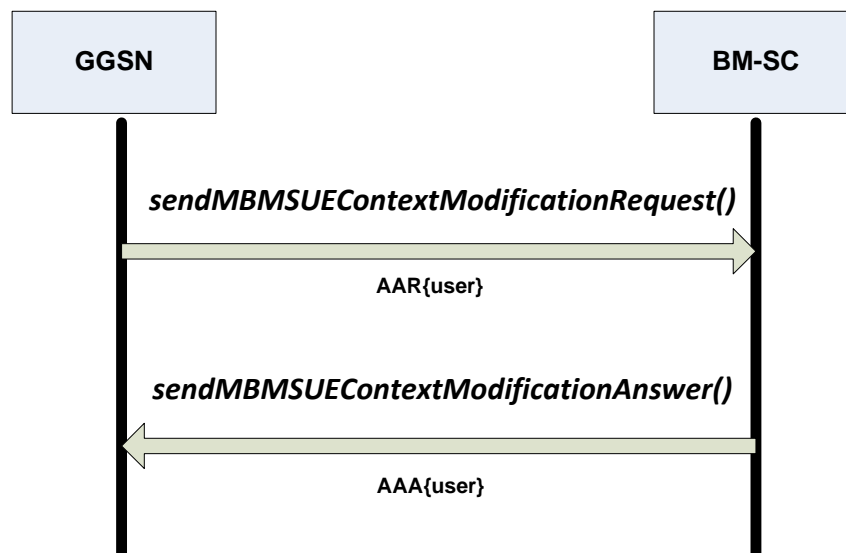


Figure 27 – MBMS UE Context Modification procedure

The GGSN sends updated MBMS UE Context parameters to BM-SC in a request message. Then, the BM-SC updates its MBMS UE Context fields and responds with an answer message, confirming the successful updating.

4.3 Gmb Messages

The Gmb interface uses not only Diameter base protocol messages (or commands), previously presented in the Diameter section, but also some messages defined in another protocol, the NASREQ. This set of messages, which are exchanged between the GGSN and the BM-SC on the Gmb interface, are presented in this section in more detail.

The messages are represented in the ABNF (Augmented Backus-Naur form) format, showing the diameter header and all relevant AVP's. There are some AVP representation rules:

- An AVP presented between "< >" must exist in the message and has a fixed position.
- An AVP presented between "{ }" must exist but can appear anywhere in the message.
- An AVP presented between "[]" is optional and can appear anywhere in the message.
- An AVP presented with a "*" may appear several times in the message.

The bold marked AVP's indicate new optional AVP's for Gmb or modified existing ones.

4.3.1 AAR Command

The AA-Request command (AAR), defined in Diameter NASREQ is sent by the GGSN to the BM-SC to request user authorization, to modify an MBMS UE Context in the BM-SC, to register the GGSN for a particular multicast MBMS bearer service or used when the GGSN needs to activate a Trace Session in the BM-SC.

Message Format:

```
<AA-Request> ::= < Diameter Header: 265, REQ, PXY >
                < Session-Id >
                { Auth-Application-Id }
                { Origin-Host }
                { Origin-Realm }
                { Destination-Realm }
                { Auth-Request-Type }
                [ Destination-Host ]
                [ Called-Station-Id ]
                [ Calling-Station-Id ]
                [ Framed-IP-Address ]
                [ Framed-IPv6-Prefix ]
                [ Framed-Interface-Id ]
                * [ Proxy-Info ]
                * [ Route-Record ]
                [ 3GPP-IMSI ]
                [ RAI ]
                [ 3GPP-IMEISV ]
                [ 3GPP-RAT-Type ]
                [ 3GPP-User-Location-Info ]
                [ 3GPP-MS-TimeZone ]
                [ Additional-MBMS-Trace-Info ]
```

4.3.2 AAA Command

The AA-Answer command (AAA), defined in Diameter NASREQ, is sent by the BM-SC to the GGSN in response to the AAR command.

Message Format:

```

<AA-Answer> ::= < Diameter Header: 265, PXY >
    < Session-Id >
    { Auth-Application-Id }
    { Origin-Host }
    { Origin-Realm }
    [ Result-Code ]
    [ Experimental-Result ]
    [ Error-Message ]
    [ Error-Reporting-Host ]
    * [ Failed-AVP ]
    * [ Proxy-Info ]
    [ Alternative-APN ]
    [ 3GPP-IMSI ]
    [ TMGI ]
    [ Required-MBMS-Bearer-Capabilities ]

```

4.3.3 STR Command

The Session-Termination-Request command, defined in the Diameter base protocol, is sent by the GGSN to the BM-SC to terminate a Diameter session. For a multicast service, this happens when the last UE Context for that bearer service is terminated. Individually, the session is terminated when the UE has requested to leave the MBMS bearer service. Also, the STR command is used to deactivate or terminate a Trace Session.

Message Format:

```

<ST-Request> ::= < Diameter Header: 275, REQ, PXY >
    < Session-Id >
    { Origin-Host }
    { Origin-Realm }
    { Destination-Realm }
    { Auth-Application-Id }
    { Termination-Cause }
    [ Destination-Host ]
    * [ Class ]
    [ Origin-State-Id ]
    * [ Proxy-Info ]
    * [ Route-Record ]
    [ Additional-MBMS-Trace-Info ]

```

4.3.4 STA Command

The Session-Termination-Answer command is sent in response to a STR command (De-registration procedure).

Message Format:

```

<ST-Answer> ::= < Diameter Header: 275, PXY >
    < Session-Id >
    { Result-Code }
    { Origin-Host }
    { Origin-Realm }
    * [ Class ]

```

```

    [ Error-Message ]
    [ Error-Reporting-Host ]
    * [ Failed-AVP ]
    [ Origin-State-Id ]
    * [ Redirect-Host ]
    [ Redirect-Host-Usage ]
    [ Redirect-Max-Cache-Time ]
    * [ Proxy-Info ]

```

4.3.5 RAR Command

The Re-Auth-Request command (RAR) is defined in the Diameter base protocol.

Message Format:

```

<RAR> ::= < Diameter Header: 258, REQ, PXY >
    < Session-Id >
    { Origin-Host }
    { Origin-Realm }
    { Destination-Realm }
    { Destination-Host }
    { Auth-Application-Id }
    { Re-Auth-Request-Type }
    [ Called-Station-Id ]
    [ Framed-IP-Address ]
    [ Framed-IPv6-Prefix ]
    [ Framed-Interface-Id ]
    [ MBMS-StartStop-Indication ]
    [ MBMS-Service-Area ]
    [ MBMS-Required-QoS ]
    [ MBMS-Session-Duration ]
    [ MBMS-Service-Type ]
    [ MBMS-Counting-Information ]
    [ MBMS-Session-Identity ]
    [ MBMS-Session-Identity-Repetition-number ]
    [ TMGI ]
    * [ 3GPP-SGSN-Address ] ; broadcast case only
    * [ 3GPP-SGSN-IPv6-Address ] ; broadcast case only
    [ MBMS-2G-3G-Indicator ]
    [ MBMS-Time-To-Data-Transfer ]
    [ Origin-State-Id ]
    * [ Proxy-Info ]
    * [ Route-Record ]

```

This command is both used to announce a session start or session stop procedure. The type of procedure is defined in the AVP *MBMS-StartStop-Indication*.

For the session start procedure, RAR is sent by the BM-SC to the GGSN's when it is ready to send data. These GGSN's have to be previously registered for the corresponding multicast MBMS bearer service. This is a request to notify interested UE's of the imminent start of the transmission. For a session stop procedure, RAR is sent by the BM-SC to the GGSN's when it considers the MBMS session to be terminated.

4.3.6 RAA Command

The Re-Auth-Answer command (RAA), also defined in the Diameter base protocol, is sent in a response to the RA.

Message Format:

```
<RAA> ::= < Diameter Header: 258, PXY >
         < Session-Id >
         { Origin-Host }
         { Origin-Realm }
         [ Result-Code ]
         [ Experimental-Result ]
         [ MBMS-StartStop-Indication ]
         [ Origin-State-Id ]
         [ Error-Message ]
         [ Error-Reporting-Host ]
         * [ Failed-AVP ]
         * [ Redirected-Host ]
         [ Redirected-Host-Usage ]
         [ Redirected-Host-Cache-Time ]
         * [ Proxy-Info ]
```

4.3.7 ASR Command

The Abort-Session-Request command (ASR), defined in the Diameter base protocol, is sent by the BM-SC to the GGSN to request that the session identified by the Session-Id be stopped.

Message Format:

```
<ASR> ::= < Diameter Header: 274, REQ, PXY >
         < Session-Id >
         { Origin-Host }
         { Origin-Realm }
         { Destination-Realm }
         { Destination-Host }
         { Auth-Application-Id }
         [ Origin-State-Id ]
         * [ Proxy-Info ]
         * [ Route-Record ]
```

4.3.8 ASA Command

The Abort-Session-Answer command (ASA), defined in the Diameter base protocol, is sent in a Response to a ASR.

Message Format:

```
<ASA> ::= < Diameter Header: 274, PXY >
         < Session-Id >
         { Result-Code }
         { Origin-Host }
         { Origin-Realm }
         [ Origin-State-Id ]
         [ Error-Message ]
         [ Error-Reporting-Host ]
         * [ Failed-AVP ]
         * [ Redirected-Host ]
         [ Redirected-Host-Usage ]
```

```
[ Redirect-Max-Cache-Time ]  
* [ Proxy-Info ]
```


5. Gmb interface implementation

After getting to know the MBMS architecture, how the BM-SC and the GGSN communicate, the protocol that supports the communication, the MBMS procedures involved and the software that supports the desired implementation, it is now the time to describe the work done throughout this project.

From all the procedures defined in the 3GPP TS 29.061 V6.9.0 [1], these are the ones involved in the scope of this project:

- Service Activation (15.5.1)
- Session Start (17.5.2)
- Session Stop (17.5.3)
- Registration (17.5.4)
- De-registration – GGSN initiated (17.5.5)
- De-registration – BM-SC initiated (17.5.6)
- Service Deactivation – GGSN initiated (17.5.7)
- Service Deactivation – BM-SC initiated (17.5.7.1)

5.1 Client/Server VS Peer-to-Peer

First of all, there must be an application that acts like a server, the BM-SC, and an application that acts like a client, the GGSN. But note that in a transport point of view, there is no server or client, because the Diameter (the supporting communication) is a peer-to-peer protocol. In a service point of view, servers can be defined to provide services, like an MBMS service session for the BM-SC, and clients to consume them, like the example of the GGSN.

5.2 Developing the Server and Client applications

Having clarified this Server-Client issue, the implementation was built with two computers connected with each other over the LAN, representing the BM-SC and the GGSN, each one of them with OpenBloX installed.

To develop an application node, a stack manager needs to be created. The stack manager is used for the purpose of stacks registrations. A stack is a class responsible for data transfer between Diameter clients and servers, handling sockets and threads.

Next, it is necessary to initialize the stack with a proper configuration file. This file has all the information regarding one Diameter node, like its IP, realm, type of node (Server or Client), supporting applications (in this case Gmb), a routing table (containing

information about the other nodes that the application will be connected with), among other important information.

OpenBloX uses XML files to provide this configuration information. As an example, parts of the configuration files are shown both for the client and server, respectively:

Diameter Client Configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<properties>
  <comment>Diameter Client configuration</comment>

  <entry key="Driver" value="com.traffix.openblox.StackImpl"/>
  <entry key="StackType" value="TYPE_CLIENT"/>
  <entry key="OwnDiameterURI" value="aaa://192.168.123.102:3867"/>
  <entry key="OwnRealm" value="av.it.pt"/>
  (...)
</properties>
```

Diameter Server Configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
<properties>
  <comment>Diameter Server configuration</comment>

  <!-- Own stack identity -->
  <entry key="OwnDiameterURI" value="aaa://
192.168.123.101:3868"/>
  <entry key="OwnDiameterIdentity" value="192.168.123.101"/>
  <entry key="OwnRealm" value="av.it.pt"/>

  (...)

  <!-- Stack/Peer configuration -->
  <entry key="Driver" value="com.traffix.openblox.StackImpl"/>
  <!-- TYPE_CLIENT or TYPE_SERVER -->
  <entry key="StackType" value="TYPE_SERVER"/>
  (...)
</properties>
```

This is the beginning of each configuration file, with information about the IP and realm and the indication of the type of node.

Proceeding, each node has the indication of the supported applications:

```
<!-- Supported applications.-->
<entry key="ApplicationIds">
  <properties>
    <!-- 3GPP vendor Id -->
    <entry key="VendorId" value="10415"/>
    <entry key="AcctApplicationId" value="0"/>
    <!-- Gmb application Id -->
    <entry key="AuthApplicationId" value="16777223"/>
  </properties>
</entry>
```

```
</entry>
```

As indicated above, both nodes use the Diameter base protocol to initiate the accounting procedure (*AcctApplicationId* = 0) and Gmb the authentication procedure (*AuthApplicationId* = 166777223). Defined by IANA [8], the Diameter base protocol is represented by the number 0 and Gmb by the number 166777223.

Client Peer table:

```
<entry key="StaticPeerTable">
  <properties>
    <entry key="PeerHost" value="192.168.123.101"/>
    <entry key="PeerPort" value="3868"/>
    <entry key="PeerRealm" value="av.it.pt"/>
    <entry key="PeerConnecting" value="true"/>
  </properties>
```

Server Peer table:

```
<!-- Statically configure peers -->
<entry key="StaticPeerTable">
  <!-- Peer of Gmb client -->
  <properties>
    <entry key="PeerHost" value="192.168.123.102"/>
    <entry key="PeerRealm" value="av.it.pt"/>
  </properties>
  (...)
</entry>
```

As previously referred, each application contains information about other nodes in peer tables, as one can see above.

After the stack initialization, the object *NetWork* must be extracted. *NetWork* represents the remote application side for the stack and manages the peer table entries. Afterwards, peer connectivity is initiated and each application is ready for receiving incoming messages.

To send a message successfully, the interested application must create a new session. To provide a better understanding of the difference between connection and session, the former is a transport level connection between two peers, used to send and receive Diameter messages, while the latter is a logical concept at the application layer and it is identified via a session Id.

Then, two types of listeners must be added to each application. The network listener, capable of handling the initial arriving request, and the session event listener, capable of managing the request's reply.

Finally, it is possible to create new messages, like an initial request, with the possibility to extend it with more AVP's. Then, these newly created messages are sent using the appropriate methods.

Client and Server applications are now implemented and, at this point, the implementation of the Gmb state machine is needed.

5.3 The concept behind State Machines

Basically, the Gmb application is an extension of the Diameter base protocol state machine. So, Gmb application should be able to:

- Recognize a set of application events, like receiving a request for registration procedure.
- Handle incoming events, like authorizing the registration request.
- Send events, like sending an answer message indicating the result of the authorization.
- Maintain all application sessions, so that the application can manage parallel sessions.

The session activities are defined by the state machine design concept:

- State – The session should contain a finite amount of states that describes the application behavior.
- Event – An event is a trigger to the session to perform normative actions that will in turn maintain the application activities.
- Action – a well defined procedure that is designated to perform a particular activity.
- Post condition – each session is activated by event, perform an action or multiple actions and modify the post condition to a new state.

The basic concept of the Gmb state machine, like other FSM, is to receive an event when currently in some state, process some pre-defined set of actions and finishing by updating the machine's state.

5.4 Managing events

At this point, it is necessary to create events defined by the type of messages received and implement the corresponding actions.

There are a set of pre-defined events, like the ones related to Capability Exchange messages, Disconnect Peer messages and Device Watchdog messages, all of these associated with de Diameter base protocol. But new events must additionally be created so they can take part of the Gmb state machine: These are the newly created ones:

IRcvSTA, IRcvSTR, RRcvSTA, RRcvSTR;

RRcvAAA, RRcvAAR, IRcvAAR, IRcvAAA;

RRcvRAA, RRcvRAR, IRcvRAR, IRcvRAA;

RRcvASA, RRcvASR, IRcvASR, IRcvASA;

These events define a reception of a type of Gmb interface message, which could be one of the following: STR, STA, AAR, AAA, RAR, RAA, ASR and ASA. The 'I' in the beginning of the events is used to represent the initiator (connecting) connection, while the 'R' is used to represent the responder (listening) connection.

Now, with pre-defined events, we must correlate them with the type of messages. This is how it is done: there is a method that analyses the command code of the received message and calls the correct function to handle it.

```
private StateEvent createStateEvent(ByteBufferWrapper bufferWrapper,
    int offset, boolean rChannel) {

    int commandCode = Header.getCommandCode(buffer, offset);
    switch (commandCode) {

        (...)

        case MessageSTAnswer.code:

            return handleST(bufferWrapper, offset, rChannel);

        case MessageAAAnswer.code:

            return handleAA(bufferWrapper, offset, rChannel);

        case MessageRAAnswer.code:

            return handleRA(bufferWrapper, offset, rChannel);
        case MessageASAnswer.code:

            return handleAS(bufferWrapper, offset, rChannel);

        (...)
    }
```

Afterwards, each corresponding function will link the events with the appropriated type of received messages. In this example, it is shown the function that manages the Session-Termination messages:

```
private StateEvent handleST(ByteBufferWrapper bufferWrapper,
    int offset, boolean rChannel) {

    return handleXX(bufferWrapper, offset, rChannel,

        PeerEvents.RRcvSTR, PeerEvents.IRcvSTR,

        PeerEvents.RRcvSTA, PeerEvents.IRcvSTA);

}
```

The next step is to match each newly defined event with the appropriate action. Each state has a processing event function that, according to the type of event, initiates the correspondent action.

```
public boolean processEvent(StateEvent event) {

    Enum type = event.getType();

    (...)

    switch (peerEvent) {

        (...)

        case IRcvAAR:

            stateMachine.unlockState();

            stateMachine.atomicActionProcessAAR(event);

            break;

        case IRcvSTA:

            stateMachine.unlockState();

            stateMachine.atomicActionProcessSTA(event);

            break;

        (...)

    }

}
```

This piece of code above represents that processing event function in the I-Open state. Thus, if a AAR message is received by the application, the *atomicActionProcessAAR()* is called. Similarly, if the received message is of type STA, *atomicActionProcessSTA()* is the called function.

5.5 Handling events with actions

One has now reached the core of the objective of this project, i.e., the development of the action methods involved in the messages transmission between the server application (BM-SC) and the client application (GGSN).

In order to avoid presenting the extensive processes and methods involved in all Gmb implemented procedures, which would be rather tiresome, only the development of the complete Service Activation procedure will be described. All the actions occurring throughout the whole activation process, the creation of the messages, the AVP's analysis involved in, together with the most relevant pieces of code implicated in this procedure will be presented, both for the server application and the client application. The remaining implemented procedures are, in a certain way, very similar to the one to be described.

In Service Activation procedure, as described in section 4.2.1, the GGSN begins with sending an AAR message requesting authorization, so the user can be able to receive data from a MBMS service.

Firstly, an AAR message needs to be created. Having done so, the relevant information is added to the message in the corresponding AVP's. Finally, the message is sent to the server. The following function is an example of what has been stated, and it is described step by step.

The following function returns a message used to request a service activation authorization. The service is defined by the parameter *service* and the requesting user by the parameter *msisdn*.

```
public static Request  
sendMBMSServiceActivationRequest_Authorization(String service, String  
msisdn)
```

The AAR message is now created, with the information of the server's realm and IP.

```
{  
  
    Request request =  
session.createRequest(MessageAARRequest.code, applicationIds[0],  
serverRealm, serverHost);
```

Next, one needs to add the Service and the MSISDN to the message AVP's. As defined in the Gmb technical specification, a request for user authorization for an MBMS bearer service is indicated by the presence of the MSISDN within the Calling-Station-Id AVP and the 3GPP-IMSI.

```
// Defining requested service
```

```
request.getAvps().addAvp(AvpFramedIPAddress.code,  
InetAddress.getByName(service).toString(), true);  
  
// Adding the MSISDN to the Calling-Station-Id AVP and the 3GPP-IMSI  
  
request.getAvps().addAvp(AvpCallingStationId.code, msisdn,  
true);  
  
request.getAvps().addAvp(Avp3GPPIMSI.code, msisdn, true);
```

At last, the message is sent over the current session.

```
session.send(request, TIME_OUT);  
  
return request;  
  
}
```

At this point, the client has successfully sent an AAR message to the server. Now, the server has to process it.

An AAR message is sent over the Gmb to indicate different purposes. It can be sent to, e.g., request user authorization for joining a service, to request user authorization for creating a UE Context or to register the GGSN for a particular service. For now, we just want it to request user authorization for joining a service.

In order to distinguish each purpose, there is a method that processes the AAR message and determines the correct intention, checking the content of the AVP's.

```
// atomicActionProcessAAR analyses the requested procedure  
public final void atomicActionProcessAAR(StateEvent event)
```

This method contains a condition that checks the CallingStationID AVP and the 3GPP-IMSI AVP and compares them. If they contain the same information, therefore we have a scenario of a Service Activation request.

```
If(avpCallingStationID.getOctetString().equals(avp3gppImsi.getOctetStr  
ing()))  
{  
    // Service Activation Procedure  
    atomicActionProcessServiceActivationRequest(event);  
}
```


The method *atomicActionProcessServiceActivationRequest()* will now check if the AAR has the purpose of user authorization for activating a service or for creating a UE Context.

In a service activation procedure, the GGSN starts by sending an authorization request for service authorization. If this authorization is granted, the BM-SC responds indicating the APN to be used for the creation of UE Context. Only then can the GGSN successfully request UE Context creation.

Thus, this method checks the AVP that is defined to contain the information about the APN. If *CalledStationId* AVP has information about the APN, we have a UE Context creation request. If not, we have an authorization for service activation.

```
// Check if the AAA is for User Authorization
// or for User Authorization/UE context creation
apn = avps.getAvp(AvpCalledStationId.code);

if (apn == null)
{
    // there is no APN information -> requesting user authorization
    (...)
else
{
    // there is APN Information -> requesting UE Context creation
    (...)
}
```

At this moment, the server has to create an answer message (AAA) with the authorization decision and, if granted, the information about the APN to use in the context creation.

```
{
    answer = request.createAnswer(ResultCode.DIAMETER_SUCCESS.code);
    logger.info("Sending AAA authorizing Service Activation...");

    // Providing the APN
    answer.getAvps().addAvp(AvpAlternativeAPN.code,
AppServer.mbmsBC.get(rService).getAPN().toString(), true);
}
```

In the code above one can see that the authorization is granted, therefore the client application sends another AAR message with the provided APN used in the creation of the context.

The server application then checks the APN and, if valid, creates the MBMS UE Context associated with the requested MBMS Service.

```
logger.info("Creating UE Context...");

// Inserting IP Multicast Address identifying MBMS BS that the UE has
joined
```

```
MbmsUeContext mbmsUEx = new MbmsUeContext (InetAddress.getByName  
(mbmsService));  
  
    // Inserting APN  
    mbmsUEx.setAPN(sApn);  
  
    // Inserting IMSI  
    mbmsUEx.setIMSI(_3gppImsi);  
    AppServer.mbmsUE.add(mbmsUEx);  
    // Inserting created MBMS UE in MBMS UE list
```

After the successful MBMS UE Context creation, a Registration procedure will take place in case the GGSN is not yet registered in the BM-SC.

Therefore, an AAR message is sent by the client application to initiate this procedure.

Consequently, the server application checks the requested service and the APN. If both are valid, it adds the identifier of the client application to the list of downstream nodes list in the requested MBMS Bearer Context.

```
//Insert in downstream nodes list  
String originHost;  
originHost = avps.getAvp(Avp.ORIGIN_HOST).getOctetString();  
AppServer.mbmsBC.get(mbmsRequestedService).addDownstreamNode(InetAddress.  
getByName(originHost));
```

At this point, the server application needs to save the session Id of the request, just so it can associate with a future De-registration request by the client application. This De-registration request, a STR command, identifies the requested service using, on its Session-Id AVP, the same value of the Session Id used to perform the Registration procedure.

```
// Save SessionID  
String sId = message.getSessionId();  
  
// Mapping Session-ID, Service and GGSN;  
SessIDServiceGGSN sessIdServiceGGSN = new  
SessIDServiceGGSN(requestedServiceIP, originHost,  
sId);AppServer.sessIdServiceGGSNlist.add(sessIdServiceGGSN);
```

Therefore, saving the information regarding the GGSN, the MBMS service and the Session Id will enable the BM-SC to easily relate the request message to the desired MBMS Bearer Service.

The server application, after sending the successful response to UE Context creation, checks if the MBMS Bearer Service is in 'Active' mode. If so, the server application initiates the Session Start procedure with the client application.

```
// Check if MBMS service is in the Active state  
if (AppServer.mbmsBC.get(i).getMBMSBearerContextState() ==  
MbmsBearerContextState.getActiveState())  
{  
    activeState = true;
```

```
}
```

```
// If the MBMS Bearer Context is in the 'Active' state,  
// the BM-SC initiates the Session Start procedure with the GGSN  
if (activeState)  
{  
    AppServer.sendMBMSSessionStartStopRequest(mbmsBearerService,  
"start");  
}
```

5.6 The implementation interface

Now that the Gmb application is finally implemented, one needs a way to easily interact with the created applications. The best approach is using a console, so input commands could be processed by the client and server applications and the results displayed.

Two console threads were implemented, each one for the server application (BM-SC) and for the client application (GGSN). Through these consoles, it is possible to send any type of requests, initiate procedures and check the content of structures like the MBMS Bearer Context and the MBMS UE Context.

Available commands for:

Server application (BM-SC) console:

```
'show services' -> Display MBMS Services  
'show service bc' -> Display Bearer Context of a MBMS Service  
'show service uec' -> Display UE Contexts for a MBMS Service  
-----  
'initiate sstart' -> Initiate Session Start Procedure for a MBMS  
Service  
'initiate sstop' -> Initiate Session Stop Procedure for a MBMS  
Service  
'initiate dreg' -> Initiate De-Registration Procedure for a MBMS  
Service  
'initiate servd' -> Initiate Multicast Service Deactivation  
Procedure for a MBMS Service  
-----  
'disc' -> Disconnect from peer  
'commands' -> Show available commands
```

Client application (GGSN) console:

```
'send rr' -> Send Registration Request  
'send dr' -> Send DeRegistration Request  
'send sar' -> Send Service Activation Request  
'send sdr' -> Send Service Deactivation Request
```

```
'send uecr' -> Send UE Context Creation Request  
'send uedr' -> Send UE Context Deletion Request  
-----  
'define sid' -> Define Session-Id  
-----  
'disc' -> Disconnect from peer  
'commands' -> Show available commands
```

6. Running the Implementation

It is now the moment to test the final implementation. To do this, two computers hosting and running the implementation are connected in the same network. Diameter messages are exchanged between the server application, representing the BM-SC, and the client application, the GGSN, and all implemented procedures are tested. The results are described and can be seen in the console of each application. Additionally, a sniffer application is running in parallel, confirming that the communication is done through Diameter messages.

BM-SC IP: 192.168.123.101

GGSN IP: 192.168.123.102

6.1 Establishing connection

This demonstration begins with the establishment of a Diameter connection between the two applications. Capability-Exchange messages are transferred between the GGSN and the BM-SC to successfully establish it.

```
INFO      [main] Loaded AVPs of Base                               [AvpFactory <init>()]
INFO      [main] Loaded AVPs of Gmb                               [AvpFactory <init>()]
INFO      [main] Loaded AVPs of NASREQ                           [AvpFactory <init>()]
INFO      [main] Loaded messages of Base                         [MessageFactoryBase addStandardBu
INFO      [main] Loaded messages of Gmb                         [MessageFactoryBase addStandardBu
INFO      [main] Loaded messages of NASREQ                      [MessageFactoryBase addStandardBu
INFO      [ConnectionsThread] Binding to /192.168.123.102:3868 0.98.2 [ConnectionsThread pre
INFO      [Thread-1] Session created                             [AppServer createSession()]

AVAILABLE COMMANDS:
'show services' -> Display MBMS Services
'show service bc' -> Display Bearer Context of a MBMS Service
'show service uec' -> Display UE Contexts for a MBMS Service
-----
'initiate sstart' -> Initiate Session Start Procedure for a MBMS Service
'initiate sstop' -> Initiate Session Stop Procedure for a MBMS Service
'initiate dreg' -> Initiate De-Registration Procedure for a MBMS Service
'initiate servd' -> Initiate Multicast Service Deactivation Procedure for a MBMS Service
-----
'disc' -> Disconnect from peer
'commands' -> Show available commands
INFO      [Thread-1] Current Session Id: 192.168.123.102;274;0 [ConsolaBMSC readConsole()]

prompt> INFO      [ConnectionsThread] Accepting /192.168.123.101:2721 [ConnectionsThread has
```

Figure 28 – Server's console

```

INFO      [main] Loaded AVPs of Base      [AvpFactory <init>()]
INFO      [main] Loaded AVPs of Gmb      [AvpFactory <init>()]
INFO      [main] Loaded AVPs of NASREQ    [AvpFactory <init>()]
INFO      [main] Loaded messages of Base  [MessageFactoryBase addStandardBud
INFO      [main] Loaded messages of Gmb   [MessageFactoryBase addStandardBud
INFO      [main] Loaded messages of NASREQ [MessageFactoryBase addStandardBud
INFO      [ConnectionsThread] Binding to /192.168.123.101:3867 0.98.2 [ConnectionsThread pre
INFO      [main] Ready                    [AppClient run()]

AVAILABLE COMMANDS:
'send rr' -> Send Registration Request
'send dr' -> Send DeRegistration Request
'send sar' -> Send Service Activation Request
'send sdr' -> Send Service Deactivation Request
'send uecr' -> Send UE Context Creation Request
'send uedr' -> Send UE Context Deletion Request
-----
'define sid' -> Define Session-Id
-----
'disc' -> Disconnect from peer
'commands' -> Show available commands
INFO      [Thread-1] Session ID: 192.168.123.101;274;0      [Console readConsole()]

```

Figure 29 - Client's console

Figure 28 and Figure 29 show the server and client consoles respectively, with all the available commands and the indication of the current session Id.

6.2 Service Activation procedure

The first described procedure is the Service Activation procedure. In this demonstration, three distinct MBMS multicast services were pre-created, and they can be presented with the command '**show services**' in the server application representing the BM-SC, as one can see in Figure 30.

```

prompt> show services

Listing available MBMS Services:
Service:          /224.1.1.1
Service:          /224.1.1.2
Service:          /224.1.1.3

```

Figure 30 - Created MBMS Bearer Contexts

Supposing that an UE is interested in receiving data from a MBMS Service identified by the multicast IP 224.1.1.2. The GGSN starts to send a request with the intention of requesting authorization from the BM-SC for the activating UE to receive data from that particular service.

In Figure 31, the GGSN sends a service activation request with the command '**send sar**' in the name of the UE identified by the IMSI "934140943". As one can see, the authorization is granted by the BM-SC along with the APN to be used for the creation of the MBMS UE context.

```

prompt> send sar
Enter MBMS Service IP: 224.1.1.2
Enter IMSI: 934140943
MBMS Service activation request sent.

INFO    [STH-Thread(192.168.123.102)] Processing AAA... [PeerStateMachineImpl atomicActionProcess
INFO    [STH-Thread(192.168.123.102)] Request successful. [PeerStateMachineImpl atomicActionProcess
INFO    [STH-Thread(192.168.123.102)] Authorization granted! Activate through APN 'APN Id2-123'
INFO    [Thread-1] Session ID: 192.168.123.101;274;2 [Console readConsole()]

```

Figure 31 – GGSN sending a service activation command

The next figure demonstrates the processing of the authorization previously granted occurred in the BM-SC. This entity gets the AAR message, verifies that is a Service Activation procedure, checks if the requested service is valid, grants/denies the authorization and answers back, providing the APN in case of granted authorization.

```

.123.101]] Processing AAR... [PeerStateMachineImpl atomicActionProcessAAR()] [18
.123.101]] Processing AA message... [PeerStateMachineImpl atomicActionProcessAA()]
.123.101]] Processing Service Activation Request... [PeerStateMachineImpl atomicAc
.123.101]] Checking for requested MBMS Service... [PeerStateMachineImpl checkMBMS3
.123.101]] Requested Service Found: MBMS Multicast Service 2 [PeerStateMachineImp
.123.101]] Processing user authorization... [PeerStateMachineImpl atomicActionPro
.123.101]] Authorization granted. [PeerStateMachineImpl atomicActionProcessService
.123.101]] Sending AAA authorizing Service Activation... [PeerStateMachineImpl ato

```

Figure 32 - The BM-SC processing the service activation request

If the GGSN indicates an invalid MBMS service (Figure 33), the BM-SC determines that there is no service with that IP (Figure 34) and informs the GGSN of the fact.

```

INFO    [Thread-1] Session ID: 192.168.123.101;274;0 [Console readConsole()]

prompt> send sar
Enter MBMS Service IP: 224.11.22.33
Enter IMSI: 934140943
MBMS Service activation request sent.

INFO    [STH-Thread(192.168.123.102)] Processing AAA... [PeerStateMachineImpl atomicActionPro
INFO    [STH-Thread(192.168.123.102)] ERROR: Requested MBMS Service not found on BM-SC [PeerS
INFO    [Thread-1] Session ID: 192.168.123.101;274;1 [Console readConsole()]

```

Figure 33 - Invalid requested service

```

.123.101]] Processing AAR... [PeerStateMachineImpl atomicActionProcessAAR()] [18:09
.123.101]] Processing AA message... [PeerStateMachineImpl atomicActionProcessAA()]
.123.101]] Processing Service Activation Request... [PeerStateMachineImpl atomicActio
.123.101]] Checking for requested MBMS Service... [PeerStateMachineImpl checkMBMSServ
.123.101]] Requested Service NOT Found [PeerStateMachineImpl checkMBMSService()] [1
.123.101]] Sending AAA message informing that requested MBMS Service was not found...
.123.101]] ----- [PeerStateMachineImpl atomicAction

```

Figure 34 - Service not found by the BM-SC

Now in order to create the MBMS UE context, the GGSN sends an AAR seeking authorization, using the command '**send uecr**', followed by the response (Figure 35).

```
INFO      [Thread-1] Session ID: 192.168.123.101;274;3      [Console readConsole()]

prompt> send uecr
Enter MBMS Service IP: 224.1.1.2
Enter IMSI: 934140943
Enter APN: APN Id2-123
MBMS UE Context creation request sent.

INFO      [STH-Thread(192.168.123.102)] Processing AAA... [PeerStateMachineImpl atomicActionPro
INFO      [STH-Thread(192.168.123.102)] Request successful. [PeerStateMachineImpl atomicActionP
INFO      [Thread-1] Session ID: 192.168.123.101;274;4      [Console readConsole()]
```

Figure 35 - The GGSN sends a create UE context authorization

The BM-SC checks the service and the APN and, as one can see in the next figure, successfully creates the UE context.

```
.123.101] Processing AAR... [PeerStateMachineImpl atomicActionProcessAAR()] [18:14:24 03/06/2007]
.123.101] Processing AA message... [PeerStateMachineImpl atomicActionProcessAA()] [18:14:24 03/06/2007]
.123.101] Processing Service Activation Request... [PeerStateMachineImpl atomicActionProcessServiceAct
.123.101] Checking for requested MBMS Service... [PeerStateMachineImpl checkMBMSService()] [18:14:24 03/06/2007]
.123.101] Requested Service Found: MBMS Multicast Service 2 [PeerStateMachineImpl checkMBMSService()]
.123.101] Checking authorization [PeerStateMachineImpl atomicActionProcessServiceActivationRequest()]
.123.101] User authorized: proceeding to UE context creation... [PeerStateMachineImpl atomicActionProce
.123.101] Creating UE Context... [PeerStateMachineImpl atomicActionProcessServiceActivationRequest()]
.123.101] UE Context created in the BM-SC: MBMS Service 224.1.1.2, APN APN Id2-123 for the UE 934140943
.123.101] Sending AAA authorizing MBMS UE context creation... [PeerStateMachineImpl atomicActionSndSer
```

Figure 36 - BM-SC creating the UE context

This UE context creation can be seen with the command '**show service uec**', as the Figure 37 demonstrates.

```
prompt> show service uec
Enter service: 224.1.1.2

SERVICE UE CONTEXTS:
Service IP Multicast Address:    /224.1.1.2
IMSI:                           934140943
APN:                            APN Id2-123
```

Figure 37 - UE Contexts stored in BM-SC

The BM-SC can inform the GGSN if an invalid APN is given in the request of a UE context creation, as one can see in Figure 38 and in Figure 39.


```

prompt> send uecr
Enter MBMS Service IP: 224.1.1.2
Enter IMSI: 934140943
Enter APN: APN Id 111
MBMS UE Context creation request sent.

INFO    [STH-Thread(192.168.123.102)] Processing AAA... [PeerStateMachineImpl atomicActionProcess
INFO    [STH-Thread(192.168.123.102)] ERROR: APN doesn't exist for requested service [PeerStateMa
INFO    [Thread-1] Session ID: 192.168.123.101;274;3 [Console readConsole()]

```

Figure 38 - Invalid APN in UE context creation request

```

.123.101]] Processing AAR... [PeerStateMachineImpl atomicActionProcessAAR()] [18:13:55
.123.101]] Processing AA message... [PeerStateMachineImpl atomicActionProcessAA()] [18:
.123.101]] Processing Service Activation Request... [PeerStateMachineImpl atomicActionPr
.123.101]] Checking for requested MBMS Service... [PeerStateMachineImpl checkMBMSService
.123.101]] Requested Service Found: MBMS Multicast Service 2 [PeerStateMachineImpl check
.123.101]] Checking authorization [PeerStateMachineImpl atomicActionProcessServiceActive
.123.101]] User authorized: proceeding to UE context creation... [PeerStateMachineImpl at
.123.101]] Sending AAA denying MBMS UE context creation (Requested APN doesn't exist)...

```

Figure 39 - BM-SC informing GGSN that the APN is invalid

Also, the BM-SC is able to deny UE context creation request to a UE that did not previously seek authorization for service activation, like the example shown in the figures below.

```

prompt> send uecr
Enter MBMS Service IP: 224.1.1.2
Enter IMSI: 123456789
Enter APN: APN Id2-123
MBMS UE Context creation request sent.

INFO    [STH-Thread(192.168.123.102)] Processing AAA... [PeerStateMachineImpl atomicAction
INFO    [Thread-1] Session ID: 192.168.123.101;274;5 [Console readConsole()]

```

Figure 40 - IMSI not authorized

```

.123.101]] Processing AAR... [PeerStateMachineImpl atomicActionProcessAAR()] [18:16:52 03/06
.123.101]] Processing AA message... [PeerStateMachineImpl atomicActionProcessAA()] [18:16:52
.123.101]] Processing Service Activation Request... [PeerStateMachineImpl atomicActionProcess
.123.101]] Checking for requested MBMS Service... [PeerStateMachineImpl checkMBMSService()]
.123.101]] Requested Service Found: MBMS Multicast Service 2 [PeerStateMachineImpl checkMBMSS
.123.101]] Checking authorization [PeerStateMachineImpl atomicActionProcessServiceActivatio
.123.101]] GGSN must request authorization for UE '123456789' for for service activation. [Pe
.123.101]] Sending AAA denying request... [PeerStateMachineImpl atomicActionSndServiceActivat

```

Figure 41 – BM-SC detecting that the IMSI is not yet authorized

6.3 Registration procedure

After a successful UE context creation, if the GGSN is not yet registered to that particular MBMS service in the BM-SC, the GGSN initiates the registration procedure.

That is exactly what is happening in the next figure.

```

INFO      [Thread-1] Session ID: 192.168.123.101;274;5      [Console readConsole()]

prompt> send rr
Enter MBMS Service IP: 224.1.1.2
Enter MBMS Service APN: APN Id2-123
MBMS Service registration request sent.

INFO      [STH-Thread(192.168.123.102)] Processing AAA...    [PeerStateMachineImpl
INFO      [STH-Thread(192.168.123.102)] Request successful.  [PeerStateMachineImpl
INFO      [Thread-1] Session ID: 192.168.123.101;274;6      [Console readConsole()]

```

Figure 42 – GGSN sending a registration request

When receiving the AAR message, the BM-SC analyses the requested service and adds the identifier of the GGSN to the list of the downstream nodes of that particular service (Figure 43).

```

.123.101]] Processing AAR... [PeerStateMachineImpl atomicActionProcessAAR()] [18:17:4
.123.101]] Processing AA message... [PeerStateMachineImpl atomicActionProcessAA()] [1
.123.101]] Processing Registration Request... [PeerStateMachineImpl atomicActionProces
.123.101]] Verifying Called Station ID AVP ... [PeerStateMachineImpl processAvpCalledS
.123.101]] Checking for requested MBMS Service... [PeerStateMachineImpl atomicActionPr
.123.101]] Requested Service Found: MBMS Multicast Service 2 through APN 'APN Id2-123'
.123.101]] GGSN Identifier: 192.168.123.101 [PeerStateMachineImpl atomicActionProcessR
.123.101]] Sending AAA message to Registration Request... [PeerStateMachineImpl atomic
.123.101]] Sending AAA message with success code... [PeerStateMachineImpl atomicAction

```

Figure 43 - BM-SC adding the GGSN to the list of downstream nodes

Meanwhile, the BM-SC stores the session Id used in this request command ("192.168.123.101;274;5") so that, in future procedures like Deregistration requests, the BM-SC can identify the service to be deregistered.

Afterwards, with the command '**show service bc**', the bearer context of the desired service is displayed, as well as the GGSN IP in the downstream nodes list parameter (Figure 44).

```

show service bc
Enter service: 224.1.1.2

MBMS BEARER CONTEXT:
Service IP Address      /224.1.1.2
APN                     APN Id2-123
State:                  pt.ptinovacao.bmsc.1
Mode:                   pt.ptinovacao.bmsc.1
Downstream nodes list: /192.168.123.101
TMGI:                   pt.ptinovacao.bmsc.1
QoS :                   pt.ptinovacao.bmsc.1
Required Capabilities:  pt.ptinovacao.bmsc.1

```

Figure 44 - Displaying the MBMS Bearer Context

6.4 Session Start procedure

To advertise the impending start of the MBMS service transmission, the BM-SC initiates the Session Start procedure, supported by the command 'initiate sstart'.

```
initiate sstart
Enter service: 224.1.1.2
INFO    [Thread-1] MBMS Service 224.1.1.2 is now in the Active state. [AppServer initiateSe
INFO    [Thread-1] Connected GGSN found in downstream nodes list. [AppServer initiateSessio
INFO    [Thread-1] Session-Id Changed to 192.168.123.101;274;5 [AppServer initiateSessionSt
INFO    [Thread-1] Sending session start procedure request... [AppServer sendMBMSSessionSta
```

Figure 45 - BM-SC initiating Session Start procedure

As one can see in the figure above, the BM-SC triggers the bearer service context to 'Active' and informs all GGSN's present in the service downstream nodes list the imminent start of data transmission. Due to the fact that the connected GGSN is present in that list, the BM-SC automatically changes the current session to the one used by the GGSN to register in the service (so that the GGSN can easily identify the service in cause) and sends the request.

When receiving the request, the GGSN correlates the session Id in the message with the service and acknowledges the session start (Figure 46).

```
prompt> INFO    [STH-Thread(192.168.123.102)] Processing Re-Auth-Request... [Pe
INFO    [STH-Thread(192.168.123.102)] Acknowledging Session Start Procedure ...
```

Figure 46 - GGSN recognizing the Session Start procedure

In the next example, the BM-SC initiates a Session Start procedure of the multicast service identified by the IP "224.1.1.1". Since the connected GGSN is not registered in this MBMS service, the BM-SC will not send any request informing the impending start of data transmission to the GGSN, as described in Figure 47.

```
initiate sstart
ervice: 224.1.1.1
[Thread-1] MBMS Service 224.1.1.1 is now in the Active state. [AppServer initiateSessionStartStop()] [18:22:43 03/06/2
[Thread-1] Session start Procedure request not sent to connected GGSN because it was not found in downstream nodes list.
[Thread-1] Current Session Id: 192.168.123.102;274;2 [ConsoleBMSC readConsole()] [18:22:44 03/06/2007]
```

Figure 47 - Request not sent to GGSN

6.5 Session Stop procedure

The BM-SC initiates the Session Stop procedure when considering the bearer service to be terminated. It turns the context state to "StandBy", checks if the connected GGSN is present in the downstream nodes list and, if so, sends a RAR message indicating the

end of the transmission with the same session Id used to register the GGSN to the MBMS bearer service.

```
INFO      [Thread-1] Current Session Id: 192.168.123.102;274;2 [ConsoleB]
prompt> initiate sstop
Enter service: 224.1.1.2
INFO      [Thread-1] MBMS Service 224.1.1.2 is now in the StandBy state.
INFO      [Thread-1] Connected GGSN found in downstream nodes list. [App]
INFO      [Thread-1] Session-Id Changed to 192.168.123.101;274;5 [AppSer]
INFO      [Thread-1] Sending session stop procedure request... [AppServe]
```

Figure 48 - BM-SC initiating Session Stop procedure

When receiving the request, the GGSN associates the session Id in the corresponding AVP with the service and acknowledges the session stop (Figure 49).

```
[STH-Thread(192.168.123.102)] Processing Re-Auth-Request... [PeerStat]
[STH-Thread(192.168.123.102)] Acknowledging Session Stop Procedure ...
```

Figure 49 - GGSN recognizing the Session Stop procedure

6.6 Service Deactivation (GGSN initiated) procedure

Now, if a UE decides to leave a particular service, the GGSN must send an STR indicating that intension.

In the next example, this request is sent with a session Id different from the one used to seek authorization for the corresponding service activation (Figure 50).

```
INFO      [Thread-1] Session ID: 192.168.123.101;274;6 [Console readConsole()] [18:17:45 03/06]
send sdr
INFO      [STH-Thread(192.168.123.102)] Processing Session Termination Answer... [PeerStateMachineImpl atomicAction]
INFO      [STH-Thread(192.168.123.102)] ERROR: Operation unsuccessful. No previous AAR corresponding to actual STR.
INFO      [Thread-1] Session ID: 192.168.123.101;274;7 [Console readConsole()] [18:26:29 03/06]
```

Figure 50 - Sending Deregistration request using a not recognized Session Id

As expected, the BM-SC could not associate the session Id indicated in the GGSN's request, so the BM-SC returns an answer indicating that fact.

```
[STH-Thread(192.168.123.101)] ERROR: Requested service not yet associated with GGSN. [PeerSta]
[STH-Thread(192.168.123.101)] Sending Session Termination Answer message ... [PeerStateMachin]
```

Figure 51 - Invalid session Id

Through the command '**define sid**', the GGSN can change the current session Id to the one used to activate the service for that IMSI (Figure 52).

```
prompt> define sid
Enter new Session-Id: 192.168.123.101;274;1
Session-Id defined.INFO      [Thread-1] Session ID: 192.168.123.101;274;1      [Console readConsole()]

prompt> send sdr
INFO      [STH-Thread(192.168.123.102)] Processing Session Termination Answer... [PeerStateMachineImpl ato
INFO      [STH-Thread(192.168.123.102)] Request successful! [PeerStateMachineImpl atomicActionProcess
INFO      [Thread-1] Session ID: 192.168.123.101;274;9      [Console readConsole()]
```

Figure 52 - Defining the new session Id and sending the request to the BM-SC

At this point, the BM-SC can identify the correct session Id and therefore acknowledge the purpose of the request.

```
[STH-Thread(192.168.123.101)] Processing Session Termination Request... [PeerStateMachineImpl ato
[STH-Thread(192.168.123.101)] Request Type: service activation authorization [PeerStateMachineImpl
[STH-Thread(192.168.123.101)] Processing Service Deactivation request (leave indication)... [Peer
[STH-Thread(192.168.123.101)] Sending Session Termination Answer message ... [PeerStateMachineImpl
```

Figure 53 - BM-SC processing the leaving procedure

In the next step, the GGSN requests the user deactivation, defining the correct session Id and sending the STR message to the BM-SC.

```
INFO      [Thread-1] Session ID: 192.168.123.101;274;9      [Console readConsole()]

prompt> define sid
Enter new Session-Id: 192.168.123.101;274;3
Session-Id defined.INFO      [Thread-1] Session ID: 192.168.123.101;274;3      [Console

prompt> send uedr
INFO      [STH-Thread(192.168.123.102)] Processing Session Termination Answer... [Pee
INFO      [STH-Thread(192.168.123.102)] Request successful! [PeerStateMachineImpl ato
INFO      [Thread-1] Session ID: 192.168.123.101;274;10      [Console readConsole()]
```

Figure 54 - GGSN sending the STR

The BM-SC then associates the session Id with the multicast service "224.1.1.2", the APN "APN Id2-123" and the IMSI "934140943", and performs the release of the UE context, together with the removal of all the authorization information regarding that user with the service (Figure 55).

```
92.168.123.101]] Processing Session Termination Request... [PeerStateMachineImpl atomicAct
92.168.123.101]] Request Type: service activation create context [PeerStateMachineImpl ato
92.168.123.101]] Processing Service Deactivation request (delete UE context indication)...
92.168.123.101]] IMSI found: 934140943 [PeerStateMachineImpl atomicActionProcessSTR()] [
92.168.123.101]] Removing Authorization information... [PeerStateMachineImpl atomicAction
92.168.123.101]] found! [PeerStateMachineImpl atomicActionProcessSTR()] [18:34
92.168.123.101]] Sending Session Termination Answer message ... [PeerStateMachineImpl ato
```

Figure 55 - BM-SC releasing the UE context

The command '**show service uec**' in the BM-SC proves the successful release of the UE context.

```
show service uec
No UE Contexts defined.
```

Figure 56 - No UE contexts defined

6.7 Service Deactivation (BM-SC initiated) procedure

The Service Deactivation procedure can also be initiated by the BM-SC. Supposing the UE of the previous procedure did not request Service Deactivation, but it is the BM-SC that actually indicates the removal of the user.

Through the server's console, a request is sent indicating that IMSI "934140943" shall be removed from the bearer service "224.1.1.2" with APN "APN Id2-123". The session Id is automatically defined to be the one used in the Service Activation procedure and the request is sent.

```
initiate servd
Enter service: 224.1.1.2
Enter APN: APN Id2-123
Enter user (IMSI): 934140943
INFO    [Thread-1] UE Context of requested IMSI found for requested service!
INFO    [Thread-1] Session Id Changed to 192.168.123.101;274;1 [AppServer init
INFO    [Thread-1] Sending Service Deactivation request... [AppServer sendMBM
```

Figure 57 - BM-SC initiating Service Deactivation procedure

Then, acknowledging the Service Deactivation, the GGSN will continue the rest of the deactivation procedure.

```
prompt> INFO    [STH-Thread(192.168.123.102)] Processing Abort-Session-Request...
INFO    [STH-Thread(192.168.123.102)] Acknowledging procedure ... [PeerStateMachin
```

Figure 58 - GGSN acknowledges the BM-SC request

6.8 De-registration (GGSN initiated) procedure

When the list of downstream nodes of the MBMS service in the GGSN becomes empty, the GGSN initiates a De-registration procedure. Using the right session Id, this procedure is initiated by the command '**send dr**' in the client application (Figure 59).

```

prompt> define sid
Enter new Session-Id: 192.168.123.101;274;5
Session-Id defined.INFO      [Thread-1] Session ID: 192.168.123.101;274;5      [Console r

prompt> send dr
INFO      [STH-Thread(192.168.123.102)] Processing Session Termination Answer... [Peers
INFO      [STH-Thread(192.168.123.102)] Request successful! [PeerStateMachineImpl atomi
INFO      [Thread-1] Session ID: 192.168.123.101;274;11      [Console readConsole()]

```

Figure 59 - GGSN initiating De-registration procedure

Receiving the Session-Termination-Request, the GGSN processes the session Id and removes the identifier of the GGSN from the list of downstream nodes of the bearer service, answering back confirming the action.

```

prompt> INFO      [STH-Thread(192.168.123.101)] Processing Session Termination Request... [Pe
INFO      [STH-Thread(192.168.123.101)] Request Type: service registration [PeerStateMachineI
INFO      [STH-Thread(192.168.123.101)] Processing service DeRegistration request... [PeerSta
INFO      [STH-Thread(192.168.123.101)] Service on hashtable: /224.1.1.2 [PeerStateMachineImp
INFO      [STH-Thread(192.168.123.101)] Checking for requested MBMS Service... [PeerStateMach
INFO      [STH-Thread(192.168.123.101)] Requested Service Found: MBMS Multicast Service 2 [Pe
INFO      [STH-Thread(192.168.123.101)] GGSN Identifier: 192.168.123.101 [PeerStateMachineImp
INFO      [STH-Thread(192.168.123.101)] Removing GGSN from the downstream nodes list... [Peer
INFO      [STH-Thread(192.168.123.101)] Sending Session Termination Answer message ... [Peers

```

Figure 60 - BM-SC removing the GGSN from the requested service

Consequently, the list of downstream nodes of the multicast service “224.1.1.2” does not contain the GGSN.

```

show service bc
Enter service: 224.1.1.2

MBMS BEARER CONTEXT:
Service IP Address      /224.1.1.2
APN                     APN Id2-123
State:                  pt.ptinovacao.bmsc.be
Mode:                   pt.ptinovacao.bmsc.be
Downstream nodes list:
                        empty
TMGI:                   pt.ptinovacao.bmsc.id
QoS :                   pt.ptinovacao.bmsc.id
Required Capabilities:  pt.ptinovacao.bmsc.be

```

Figure 61 - Bearer context of the requested service

6.9 De-registration (BM-SC initiated) procedure

This procedure can also be initiated by the BM-SC when the specific bearer service is terminated. Supposing the previously GGSN is still on the bearer service downstream nodes list, the BM-SC sends a request indicating the service termination.


```
INFO [Thread-1] Current Session Id: 192.168.123.102;274;1 [ConsoleBMSC readConsole()]
prompt> initiate dreg
Enter service: 224.1.1.2
INFO [Thread-1] Initiating De-registration procedure. [AppServer initiateDeRegistrationProcedur
INFO [Thread-1] Connected GGSN found in downstream nodes list. [AppServer initiateDeRegistrati
INFO [Thread-1] Session Id Changed to 192.168.123.101;274;2 [AppServer initiateDeRegistrationP
INFO [Thread-1] Sending service de-registration request... [AppServer sendMBMSDeRegistrationPr
```

Figure 62 - BM-SC sending a service termination indication

Due to the fact that the connected GGSN is in the service list, the GGSN receives the request and answers back confirming the termination (Figure 63).

```
INFO [STH-Thread(192.168.123.102)] Processing Abort-Session-Request... [PeerState
INFO [STH-Thread(192.168.123.102)] Acknowledging procedure ... [PeerStateMachine
```

Figure 63 - GGSN acknowledging the procedure

When receiving the answer, the BM-SC releases all the UE contexts, along with all authorization information regarding the service and the GGSN. Finally, the GGSN is removed from the list of downstream nodes list (Figure 64).

```
[STH-Thread(192.168.123.101)] Processing Abort-Session-Answer... [PeerStateMachineImpl atomicActionProcessA
[STH-Thread(192.168.123.101)] Success: GGSN 192.168.123.101 acknowledged Session De-registration procedure!
[STH-Thread(192.168.123.101)] Releasing MBMS UE Contexts... [PeerStateMachineImpl atomicActionProcessASA()]
[STH-Thread(192.168.123.101)] Releasing UE Context of IMSI 934140943 with APN APN Id2-123 [PeerStateMachine
[STH-Thread(192.168.123.101)] Removing Authorization information of all UE's of requested MBMS service... [
[STH-Thread(192.168.123.101)] Service: /224.1.1.2 [PeerStateMachineImpl atomicActionProcessASA()] [19:07:1
[STH-Thread(192.168.123.101)] Found authorization information: removing it. [PeerStateMachineImpl atomicAct
[STH-Thread(192.168.123.101)] Removing GGSN 192.168.123.101 from downstream nodes list... [PeerStateMachine
```

Figure 64 - BM-SC removing all information regarding the GGSN and the service

6.10 Closing connection

In case no more procedures need to be initiated, the connection between the GGSN and BM-SC is closed.

```
disc
Disconnecting from peer...
Exiting console...
```

Figure 65 - Disconnecting

7. MBMS Evolution

Nowadays there is an intense competition in the world of the multimedia content delivery in cellular networks. Being an enhancement to the existing network infrastructure of 2G and 3G networks, MBMS has the big advantage of being easily integrated into existing UMTS networks, contrasting with other similar technologies like DVB-H, MediaFLO and DMB. Despite this, it is important that MBMS design continues to evolve in order to maintain the competitiveness of Multicast and Broadcast systems.

It is essential to optimize the MBMS design limitations, such as cell edge performance, memory requirements, priorities regarding the same MBMS service, among other limitations. However, these optimizations of technical aspects should not be the only focus for the enhancement of MBMS systems: the evolution must also follow the behavior and preferences of users that are being provided with currently deployed unicast services, such as Mobile TV. This user observation will provide valuable information for the design of evolved applications, content and network infrastructure. Then, the final system will have more chances to become a successful product, as it will better respond to and anticipate customer needs.

The reach for an improved delivery of multimedia content is still under investigation and, in order to guarantee an efficient usage of network resources, sharing channels among many user equipments is one of the privileged aspects regarding network optimization. Moreover, multi-resolution schemes can bring an enhancement on the scalability in UMTS environments.

In the next sub-sections, several state-of-the-art enhancements to the 3GPP MBMS for current and future systems will be described. These enhancements comprise multi-resolution schemes, together with advanced receiver concepts for exploiting their full potential and hierarchical modulation.

One of these enhancements, the IMS-MBMS integration proposed by C-MOBILE [17][18][19][20], will have a particular focus in Chapter 8.

7.1 MIMO Antennas

During broadcast and multicast transmissions in a mobile cellular network, some receivers will get better signal to noise ratios than others. Therefore, the capacity of the communication link for these users is higher. In broadcast transmissions, some capacity can be exchanged between good and bad communication links and the trade-off can be worthwhile. Using multiple antennas both at transmitter and receiver is one possible method to improve the efficiency of the WCDMA network.

MIMO (Multiple Input Multiple Output) is a multi-antenna topology that pushes the capacity and throughput limits as high as possible without an increase in spectrum bandwidth. Multiple antennas are used at the transmitter and receiver so that the signals at the two ends are combined in such way that the quality and data rate of each communication will be improved. The next figure illustrates the different types of multi-antenna topology, including SISO (Single Input Single Output), MISO (Multiple Input Single Output), SIMO (Single Input Multiple Output) and MIMO:

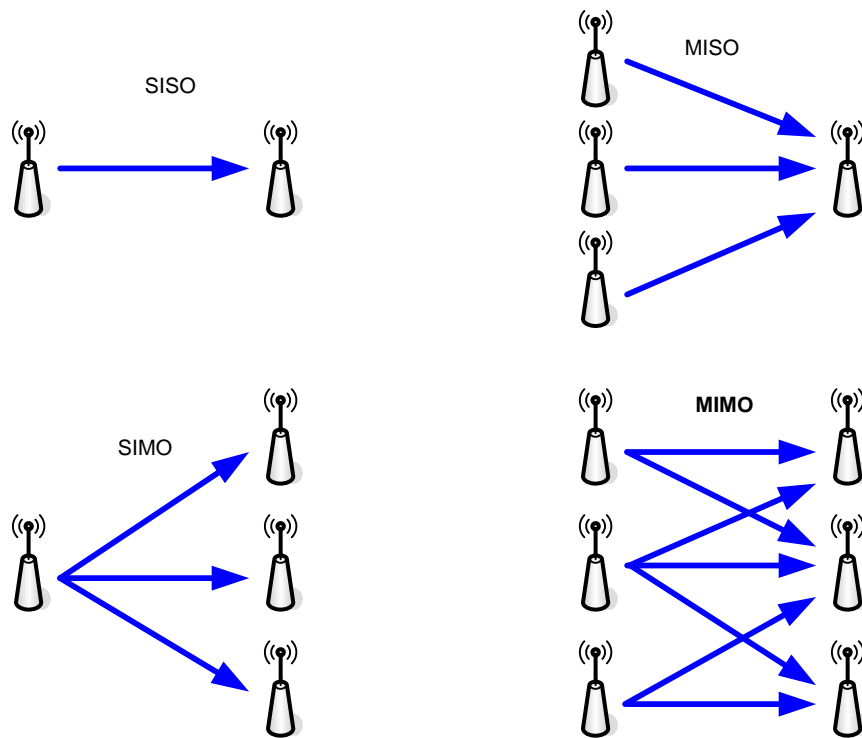


Figure 66 - Different configurations of multi-antenna

The capacity of a MIMO system increases linearly with the number of antennas in the presence of a scattering-rich environment and such an advantage can be used to increase both the network's quality of service and the operator's revenues significantly. MIMO can be seen as an extension of the old time-distributed smart antennas technology, where time is complemented with the spatial dimension inherent in the use of multiple spatially distributed antennas.

In order to create independent transmission paths, receive and transmit antennas must be sufficiently separated in space and in polarization, allowing the individual information carried in each transmission to be recombined at the receiver and, therefore, data rate is enhanced.

MIMO is currently being adopted as an enabling technology for the new mobile and wireless standards. It is considered fundamental in modern digital communication to provide substantial capacity increments and has the potential to address the unprecedented demand for wireless services, not only for unicast services but also for broadcast/multicast services, like is the case of MBMS.

7.2 Dual receiver

Dual receiver is a method for communicating between a network and a UE. It consists of the UE receiving a first signalling from a first network node at a first frequency. Next, the UE will receive a second signalling via a point-to-multipoint control channel from a second network node at a second frequency. Later, the UE will receive a request (carried on the point-to-multipoint control channel) from the second network node at the second frequency and, finally, the UE transmits to the first network node the response to the request of the second node. This described procedure is an example of a patent [9] based on the dual receiver technology and illustrated in the next figure:

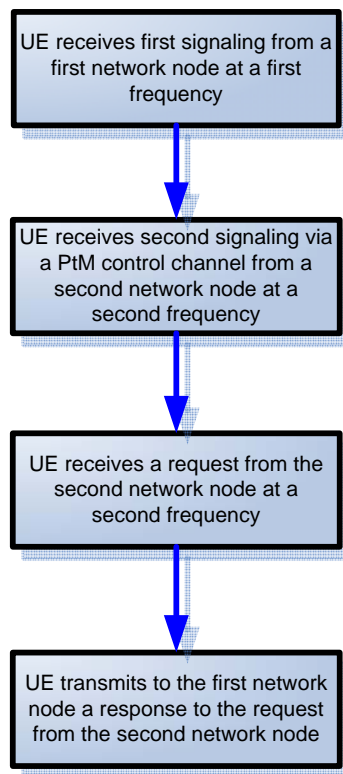


Figure 67 - Dual receiver process example

In a MBMS scenario, UEs with dual receiver should be capable of simultaneously receiving dedicated services and MBMS point-to-point/point-to-multipoint services on one frequency and MBMS services sent on point-to-multipoint bearers on another frequency. The first one, could be FDD or TDD, is the frequency on which the UE is camping for reception of dedicated services. In the other frequency, also FDD or TDD, the UEs will receive the dedicated MBMS services with the additional MBMS capabilities.

This dual receiver method can provide benefits to broadcast/multicast technologies and currently MBMS is evolving through the specification of the behavior of UEs with a dual receiver, allowing to increase the throughput in a MBMS architecture.

7.3 MBMS on HSDPA (Adaptive Coding and Modulation)

High-Speed Downlink Packet Access (HSDPA), standardized for UMTS, a third generation mobile telephony communications protocol in the High-Speed Packet Access (HSPA) family, provides up to 12.8 Mbps on a 5 MHz carrier, but only for best effort packet data services in the downlink (further speed increases are planned for the near future). QPSK and 16QAM modulation schemes are used in an adaptive manner depending on the channel state.

HSDPA, besides of providing great user peak data rates, high throughput and reduced delays, can also improve the spectral efficiency of asymmetrical downlink and bursty packet data services. All these factors are accomplished by using several techniques that rely on the transfer of some medium access control functionalities from the RNC to the Node B and on the implementation of a new type of transport channel, called High-speed downlink shared channel (HS-DSCH). This new extension of the WCDMA specification allows a certain amount of channelization codes and transmission power in a cell to be dynamically shared among other users.

One of these abovementioned techniques is the Fast Link Adaptation that makes use of the Adaptive Modulation and Coding (AMC). Instead of using power control to compensate the variations of downlink radio conditions, the transmitted power is kept constant and, for every Transmission Time Interval (2ms in HSDPA) for each UE, the modulation and coding of the transport block is chosen. The link can employ QPSK for noisy channels and 16QAM for clearer channels. The former, besides having twice higher bit rate, penalizes users on throughput and is more prone to errors due to interference and noise. The latter can tolerate higher levels of interference and maximizes the throughput.

Adaptive modulation and coding require channel information that can be directly measured at the receiver through the channel quality indicator feedback from the UE. By exploiting this channel information, adaptive modulation systems improve the rate of transmission for actual channel conditions and optimize bit error rates.

Due to its point-to-point nature, HS-DSCH is currently used for unicast services, but the rich features, such as the Adaptive Modulation and Coding, make HS-DSCH a good enhancement for broadcast/multicast services.

There are several advantages when transmitting MBMS data over HSDPA channels, like the better efficiency of radio resources by using HS-DSCH for data transmission, easier simultaneous reception of MBMS/non-MBMS services and higher user bitrates.

Therefore, data transmission over HSDPA channels has been added in 3GPP Rel-5, thus supporting point-to-point or point-to-multipoint MBMS transmissions.

7.4 Single Frequency Network (SFN)

Another major potential improvement for MBMS is the use of Single Frequency Network (SFN) operation of MBMS on the physical layer.

One way to increase MBMS coverage and to improve the radio link performance is the use of macro-diversity i.e., using several receiver and/or transmitter antennas for transferring the same signal. Being MBMS broadcasted simultaneously in different cells, network resources are not consumed when macro-diversity for MBMS is employed (this scenario does not occur when using a dedicated physical channel).

Macro-diversity objective is to supply diversity in situations where the UE is far from the base stations, in order to compensate the transmission path loss of a terminal located at the edge of the cell or to increase the network capacity by reducing the amount of transmit power needed to reach a distant receiver.

There are two types of broadcasting networks: the Multi-Frequency Networks (MFN) and the Single-Frequency Networks (SFN). The former operates several transmitters on a number of different frequencies while the transmitters in the latter send simultaneously the same signal over the same frequency channel. Using the same frequency, synchronization of the different base stations becomes more accurate, the coverage area increases and the outage probability decreases. In downlink, the combining takes place at the mobile, which has to demodulate and then combine the signals received from the different base stations in the active set.

In SFN the receivers and transmitters are usually synchronized with the others, using GPS or a signal from the network as a reference clock. The transmission on the same time-frequency resource initiated by the Node Bs can be possible because the signals will superpose within the cyclic prefix and thus the terminal can employ a single receiver to demodulate the superimposed signals. If we are dealing with MFN, separated receiver chains will be needed to demodulate the signals in the UE if there is no synchronization between the base stations.

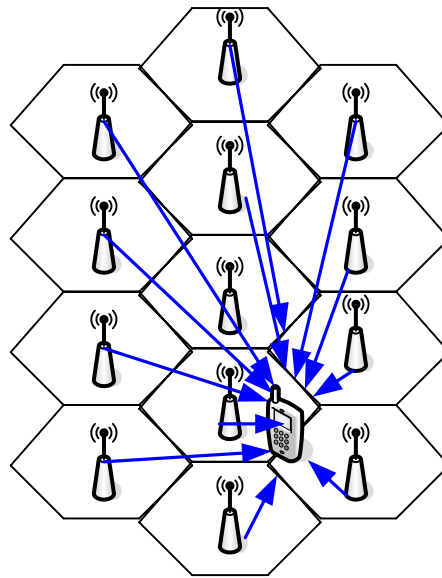


Figure 68 – Multicast/Broadcast SFN area

All the advantages that SFN can bring are seriously considered by 3GPP and Multicast/Broadcast Single Frequency Network (MBSFN) is a concept that promises a considerable increase in spectral efficiency for MBMS. Inter-cell-interference can be drastically reduced and the number of broadcasted channels boosted if unicast and multicast transmissions are separated in frequency and cells are grouped into MBMS clusters, where all the cells in each cluster use the same scrambling code.

7.5 Orthogonal Frequency-Division Multiple Access (OFDMA) Technology

The OFDM (Orthogonal Frequency-Division Multiplexing), a widespread technology in many wireless markets, sub-divides a single, high-speed data signal into several lower speed signals for parallel transmissions that are then transmitted simultaneously at different frequencies to the receiver, creating a system with increased tolerance to noise and multi-path, while simultaneously enabling more efficient use of bandwidth allocation. Moreover, a multiplicity of high-speed channels over a single frequency band is enabled through a more efficient use of bandwidth.

In the multi-user version, the Orthogonal Frequency-Division Multiple Access (OFDMA), both time and/or frequency resources are used to separate the signal into sub-channels, being each sub-channel allocated to a different user, in view to support differentiated Quality of Service, i.e. to control the data rate and error probability individually for each user. This happens because in OFDMA each user can be treated separately independent of the distance from the base station, location, power requirements and interference.

Typically, OFDMA burst consist of several OFDM symbols. The sub-channels and the OFDM symbol period are the finest allocation in the frequency and in time domain, respectively. Therefore, each user has is own slot in time and in frequency. In the next figure sub-channels are represented by arrows (each one allocated to a user) and the lines shown at different times represent the different OFDM symbols.

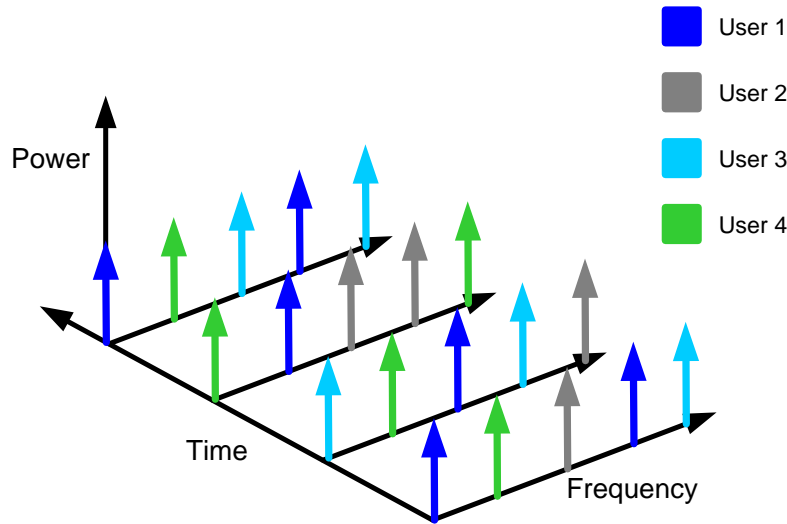


Figure 69 - Example allocation of user resources in OFDMA

OFDMA technology enables more efficient duplexing techniques, such as FDD (Frequency Division Duplexing) and TDD (Time Division Duplexing), creating a signal that is immune to interference and capable of high data throughput. Moreover, the technology improves power gain per channel, resulting in better coverage and range availability, larger cell radius, smaller power amplifiers and antennas and subsequently, lower investment in capital equipment. Therefore, OFDMA is considered to be well suited for MBMS downlink transmission.

8. IMS-MBMS integration

8.1 Introduction

It is a fact that evolution of mobile communications is advancing into a more complex multimedia service platform. Currently, there are several projects working on an approach to bring a generic multimedia streaming subsystem that address personalization and context-awareness solutions.

C-MOBILE, one of these projects, is a European IST project that works on enhancing MBMS for systems beyond 3G at Radio, RAN and Core Network levels. This project has the long-term objective of delivering multicast/broadcast services across multiple transport bearers within a converged global network architecture context.

Clearly, the main objective of C-MOBILE is to foster the evolution of the mobile broadcast business by providing enhancements to the 3GPP MBMS for systems beyond 3G. Since MBMS is all about multimedia content, the interaction with the 3GPP IP Multimedia Subsystem (IMS) can bring an extreme relevance to future mobile environments.

8.2 Integration motivations

In order to deliver multimedia services, such as streaming, content download or video telephony, just to name a few, IP protocol has been widely adopted on both application and networking level and IMS was the chosen architecture to deal with multimedia content over mobile networks.

IMS and MBMS can be integrated with the aim of supporting MBMS in a converged architecture. Such integration would be beneficial since multicast services would be controlled by IMS entities and both content providers and IMS users would be able to send multimedia content to user groups preserving network resources. Thus, considering IMS as the emerging technology for fixed and mobile convergence, the main requirement is to enable the delivery of multimedia content to a group of IMS users using a multicast-broadcast bearer wherever possible.

However, MBMS is not integrated in the future IMS based control architecture because IMS currently supports only unicast services. Thus, possible integration remains an attractive area of research. As a matter of fact, 3GPP has started work on standardizing the integration of MBMS under IMS for Release 7.

8.3 IMS architecture

The IP Multimedia Subsystem (IMS) is an architectural framework for delivering internet protocol (IP) multimedia to mobile users. 3GPP Release 5 brought IMS into play as an approach to delivering multimedia services over GPRS. Later, 3GPP, 3GPP2 and TISPAN updated this approach by requiring support of networks other than GPRS, such as CDMA2000, Wireless LAN and fixed line.

Since IMS is aware of the service requested by the user, it has the advantage of providing QoS in the user plane and to enable the operators to improve their charging schemes for multimedia sessions. Additionally, IMS allows the integration of different services offering improved as well as additional services to customers. This service awareness is mainly due to the Session Initiation Protocol (SIP) signalling, defined by the Internet Engineering Task Force (IETF), for managing the real-time multimedia streams.

As one can see in the diagram below, IMS architecture allows the user to access to multimedia services from various devices using the IP protocol or traditional telephony systems.

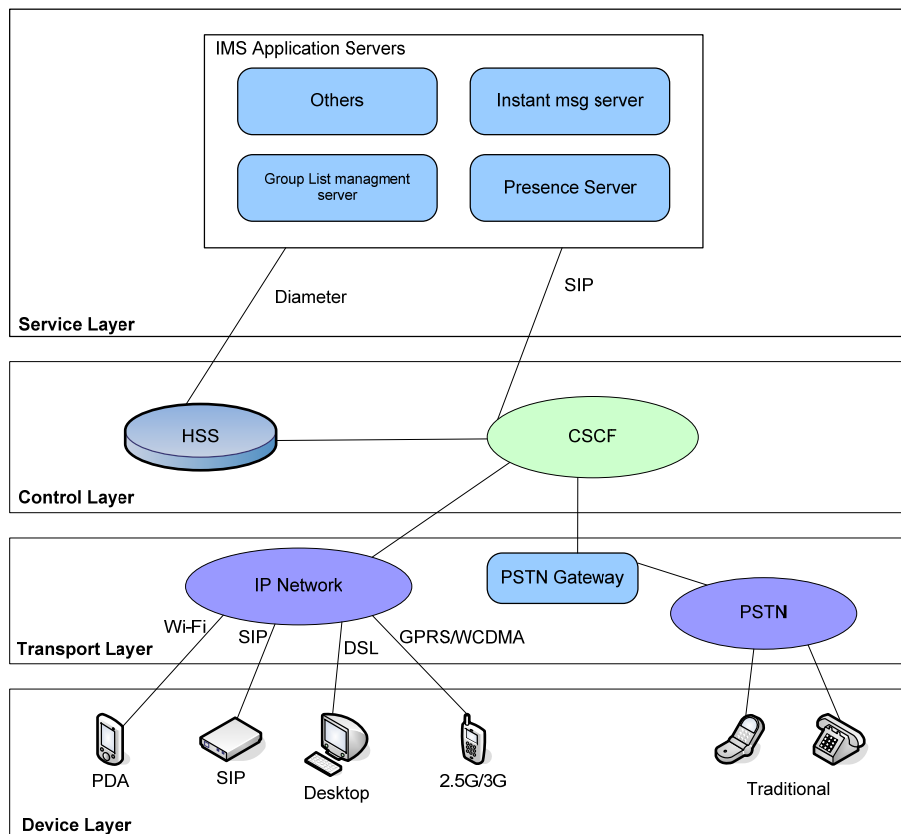


Figure 70 - IMS Architecture

This architecture is divided in four layers:

- **Device layer:** IMS offers many choices of terminals for the user to access multimedia content, such as computers, PDAs, mobile phones. Also, traditional analog equipments are able to connect to the network infrastructure, although not being able to connect to the IP network.

- Transport layer: this layer is responsible of handling SIP sessions and providing the conversion of data between analog and digital formats. Wi-Fi, SIP, DSL, GPRS and WCDMA are a variety of transmission media supported by IMS when establishing connections between the end-point devices and the IP network. Although not having an IP nature, traditional equipments are able to make and receive calls thanks to PSTN Gateway.
- Control layer: One of the core elements of the control layer is the Call Session Control Function (CSCF). This element handles SIP registrations and processes SIP signal messaging of the appropriate Application Server in the Service layer. Other element, Home Subscriber Service (HSS) is responsible of storing all the information regarding the service profile of each end-point user.
- Service layer: The three layers below Service layer provide a standardized and integrated platform that allow service providers to deliver a variety of multimedia services, all hosted by Application Servers. Each AP may host multiple services and this flexibility reduces the workload in the Control layer.

However, the design of IMS is not optimised for the delivery of files or other type of non real-time formats. The other problem with IMS is the impossibility to support point-to-multipoint connections and, consequently, network resources are consumed redundantly in scenarios of services involving groups of users receiving the same information.

Nevertheless, by creating a common multicast bearer connecting the media server to all the participants, both radio and core network resources are saved, thus becoming available for other services.

8.4 MBMS-IMS integration analysis

Possible synergies between IMS and MBMS are apparent: MBMS is used to deliver both real-time and non real-time multimedia content through broadcast and multicast content while IMS is being used for session control through SIP signalling. Gathering these characteristics in one common architecture, i.e. by having multicast services being managed by IMS entities, both IMS users and content providers could transmit multimedia content to user groups and, as a result, network resources are saved.

However, MBMS Release 6 does not take into consideration the function of IMS in the core network and its importance of providing a single platform of multimedia content delivery based on IP technology. Hence, it would be essential to investigate how MBMS could make the best use of IMS capabilities, how the multimedia delivery services could be integrated within the IP system and finally how the BM-SC should be developed and evolved to be integrated into a converged global multicast broadcast system.

Up to now, IMS and MBMS are separated subsystems sharing no common interfaces in order to utilize each other. The achievement of an integrated network architecture results in new interfaces, modified protocols and enhanced functionalities.

8.5 The two candidate architecture approaches

Two possible architecture approaches were considered to be the base of the integration of IMS and MBMS technologies: the first one pinpointed to an independent BM-SC being part of an integrated architecture and the second one all BM-SC functional elements would be distributed across various IMS entities.

8.5.1 IMS-MBMS integration proposal for 3GPP R7/R8 Network

Best appropriated for a 3rd Generation Partnership Project Release 7/8 (3GPP R7/R8) architecture, this integration proposal is the one in which IMS applications can use MBMS with the BM-SC being left unchanged, i.e. preserving its functionality and structure as possible.

With this approach, both MBMS and IMS systems are preserved as much as possible, except for some slight enhancements. In this scenario IMS is used to control MBMS bearers and to acknowledge the needs for broadcast/multicast bearer support. Content providers can initiate a MBMS service by directly connecting to the BM-SC or indirectly through an IMS Application Server.

The figure below shows a 3GPP approach of this architecture:

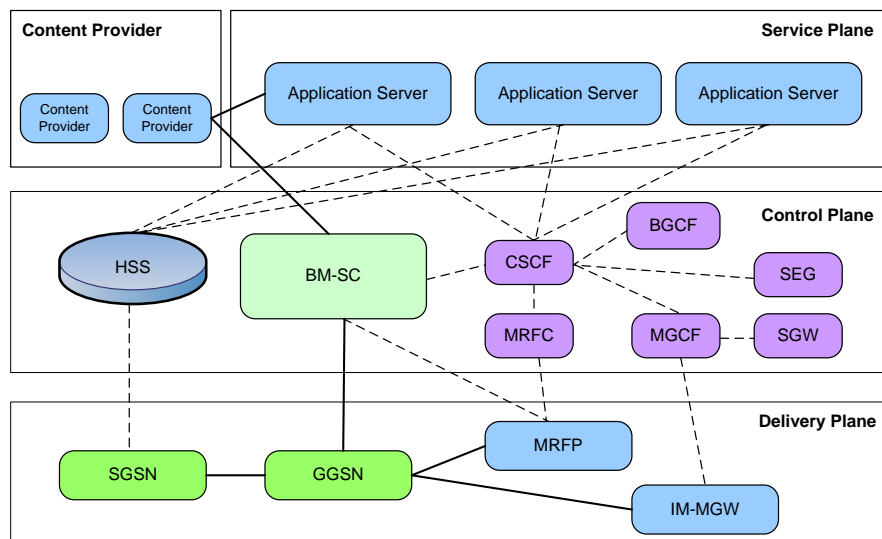


Figure 71 - R7/R8 IMS-MBMS Integration Architecture

Although the fact that there are no new network entities, there is the need to change the existing entities and to implement new interfaces in order to have communications between IMS elements and BM-SC.

From the R7/R8 architecture, C-MOBILE had come with a network entity design that consists in a five-layered framework that is presented in the figure below.

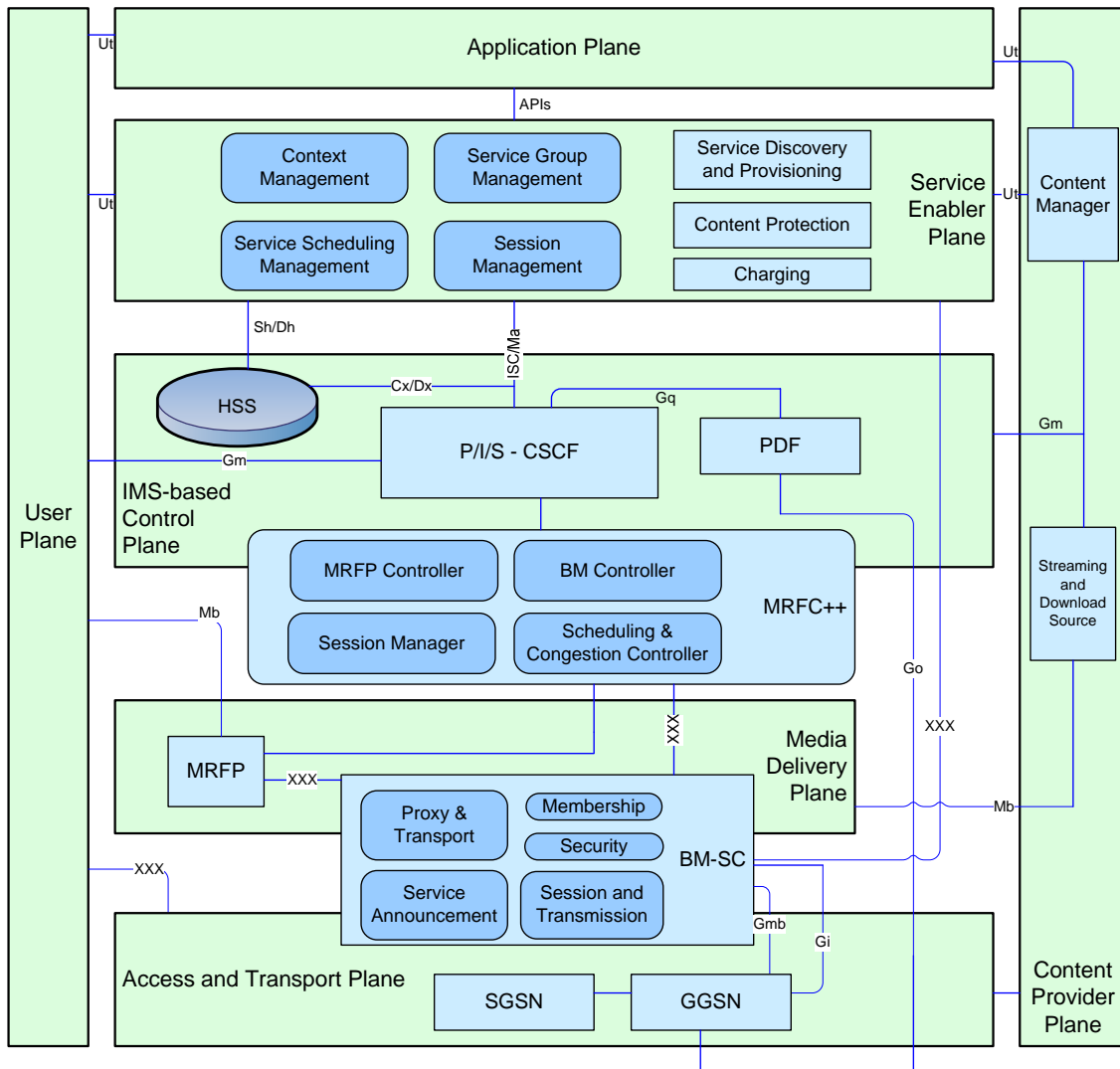


Figure 72 - R7/R8 IMS-MBMS Network Entity Design

In the Access and Transport Plane, BM-SC is providing functions for MBMS user services provisioning and delivery. It works as an entry point for MBMS transmission coming from the content provider, authorizes and initiates MBMS Bearer Services and is responsible for scheduling and delivering MBMS transmissions. As shown in the network entity design figure, all MBMS R6 BM-SC sub-functions (Membership Function, Session and Transmission Function, Proxy and Transport Function, Service Announcement Function and Security Function) are integrated in the BM-SC entity.

Regarding the Media Delivery Plane, there is the need to define an interface between MRFP and BM-SC. MRFP can be considered to be a content provider when, for instance, a multiparty conference is taking place.

In IMS-Based Control Plane, there is a functional block named Broadcast/Multicast Controller that handles data and signalling between MBMS and IMS functional entities. For signalling transmission, BM-SC connects directly with the IMS control layer and, for data purposes, BM-SC controller interacts with the BM-SC through IMS MRFP.

Value-added services are provided by several Application Servers within the Service Enabler Plane. These ASs can also be responsible for controlling the sessions between BM-SC and the Content Provider.

8.5.2 IMS-MBMS integration proposal for an evolved/Converged Network

The second approach considers the functions of the BM-SC to be distributed across the various IMS entities and satisfies the long-term vision of the 3GPP Service Architecture Evolution/Long Term Evolution (SAE/LTE) architecture.

To come with a possible architecture integrating IMS with MBMS over an evolved network, C-MOBILE considered the work on SAE/ LTE that is on-going within 3GPP and further a work based in a next generation IP-based converged network under Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN). Both TISPAN and 3GPP consider IMS as the fundamental architecture layout for the evolved service architecture for MBMS.

The next figure presents the evolved IMS-MBMS architecture, where existing network entities are enhanced and LTE/SAE entities are introduced.

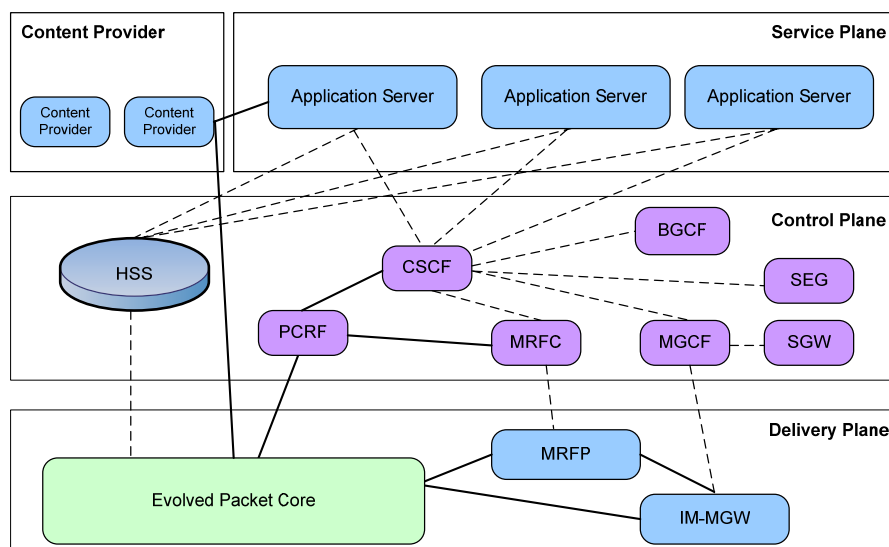


Figure 73 - Evolved IMS-MBMS Integration Architecture

New Applications Servers are now introduced and some changes have to be done in the architecture in order to support multicast media delivery from the Content Providers to the terminal through the core network. Thus, the core routers on the IP layer should handle multicast addresses.

Enhancements to support broadcast/multicast transmission in the network result in a higher level of entities complexity. Yet, the design of the evolved architecture remains more or less consistent since, apart from the AS, no new entities are introduced. Some MBMS R6 BM-SC functionalities, such as service announcement, group management security and service discovery, are moved to the Service Enabler Layer, as illustrated in the next figure.

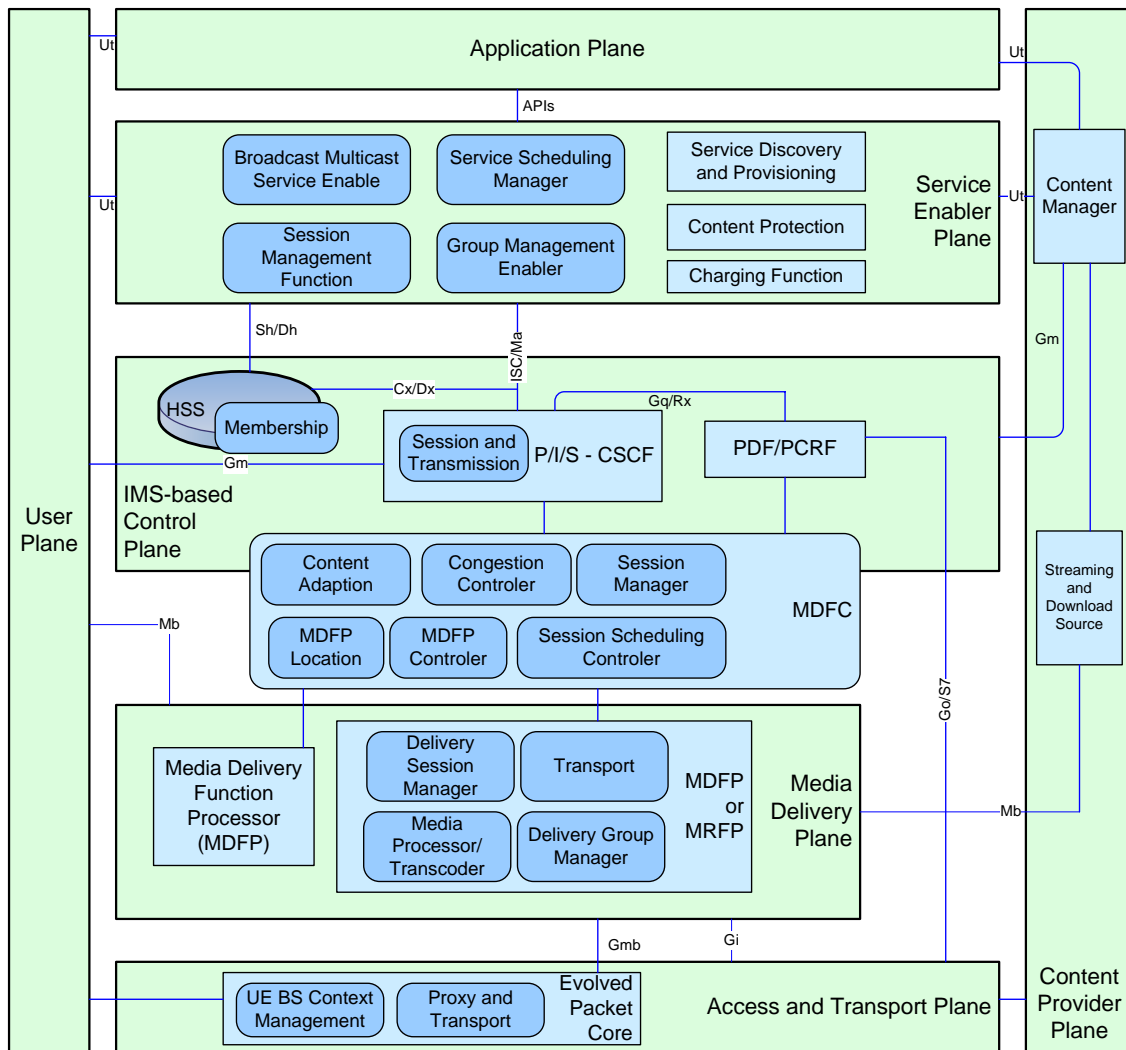


Figure 74 - Evolved Integration Network Entity Design

Having this network design, BM-SC functionalities UE BS Context Management and Proxy & Transport are contained in the packet core at the Access and Delivery Plane contains, yet being modified.

Transport and pre-processing of content take place in the Media Delivery Plane. Media Delivery Function Processor (MDFP) is placed between the Content Provider and the Evolved Packet Core as a gateway. This entity, together with MDFC, performs ciphering of media, error correction coding, transcoding and mixing of different media. Proxy and Transport, a BM-SC functionality, takes place in MDFP/MRFP.

IMS-Based Control Plane consists of HSS, CSCFs, PCRF and MDFC. These entities are short-described below:

- **HSS:** the central database in the IMS network. No new enhancements are required in this integrated architecture.

- **CSCF**: this entity is responsible of transferring the user profile, containing the subscriber service groups, from the HSS when the user establishes a connection to the IMS. Afterwards, service triggers are executed and SIP messages sent to the group server, resulting in the desired service activation.
- **PCRF**: an extension of IMS PDF, this component is responsible for QoS aspects, policy management and charging functions.
- **MDFC**: provides multi-user services, announcements or media transcoding. Furthermore, this entity is responsible for multicast addresses reservation and administration.

The Broadcast-Multicast Service Enabler (BM-SE), present in the Service Enabler Plane, provides some BM-SC Release 6 functions, such as security management functions, service description and service guide aggregation (broadcast/multicast), content scheduling, statistics collection for content delivery, quality of experience statistics collection and group management.

8.6 The Final Architecture

After a thorough investigation and analysis of the two possible IMS-MBMS integration architectures, C-MOBILE considered the significantly impact and ample innovation scope of the IMS-MBMS integration for 3GPP SAE/LTE architecture. Therefore, the new evolved converged network is defined so that there is no longer a single BM-SC, but distributed BM-SC functionalities among several IMS entities.

Based on the network entity design of the chosen architecture, and with a better understanding of the implications of that architecture, some fundamental changes, shown in the next figure, were made.

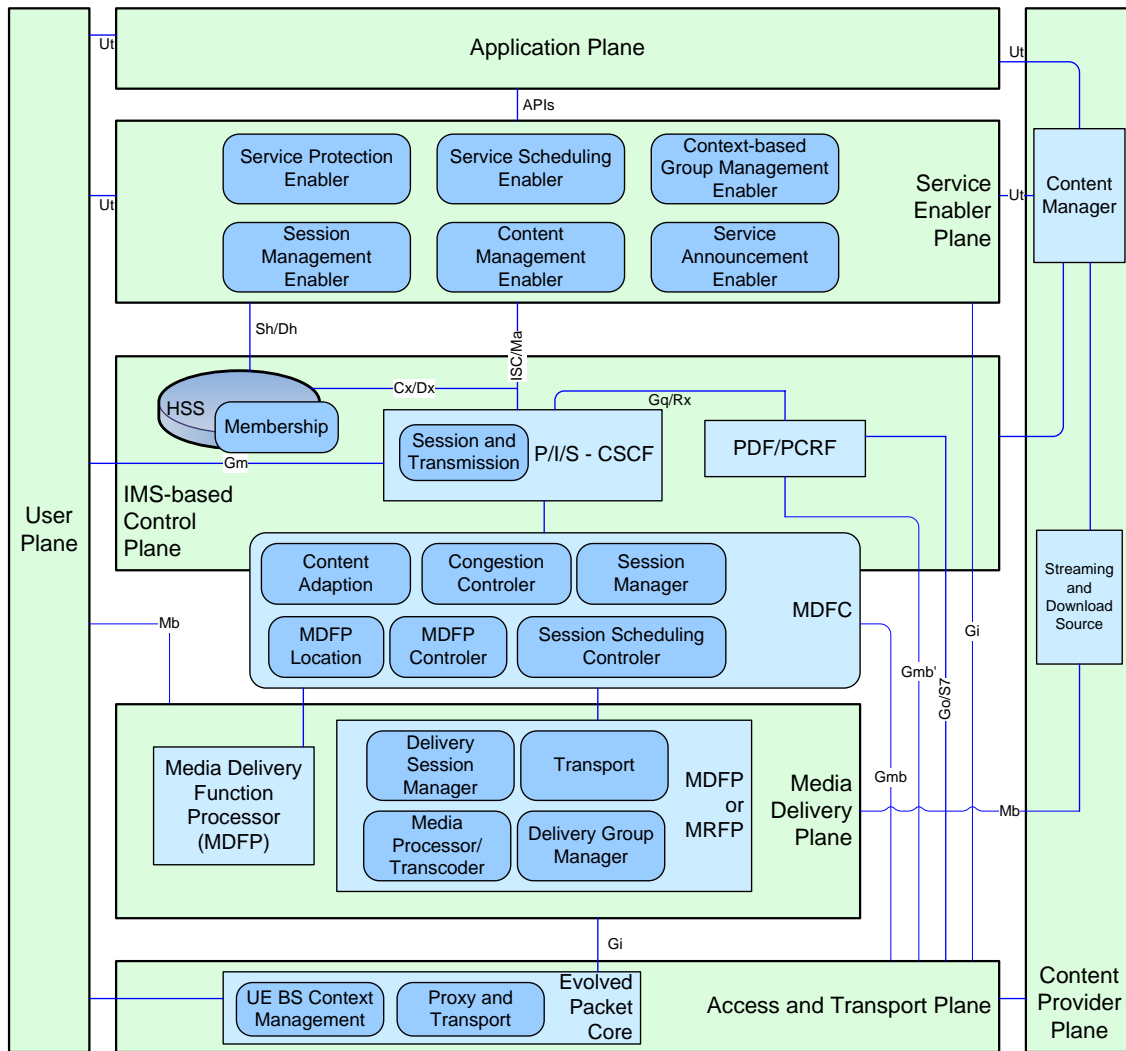


Figure 75 – IMS-MBMS Network entity Design

With the aim of increasing flexibility and to allow a better separation of functionality along the re-usability of existing functional elements, the Broadcast Multicast Service Enabler (BM-SE) was dissolved into multiple service enablers.

8.6.1 Key Service Enablers

Service enablers are components that perform a set of generic control functions to act as a bridge between the content provider and content consumer. These are the four key service enablers that support the fundamental MBMS user service provisioning and delivery functions, once supported by the MBMS BM-SC R6:

- **Service Protection Enabler (SPE):** limits the access to multicast service transmissions so that only registered or group-subscribed User Equipments can access the services.

- **Service Announcement Enabler (SAE):** responsible for announcing all available services and its related information to the end-user.
- **Session Management Enabler (SME):** handles all the signalling flow between the Service Enabler Plane and the user, such as Session Start and Session Stop procedures.
- **Context-based Group Management Enabler (CGME):** enabler where the creation, deletion and management of user groups occurs.

Additionally, two enablers support the previously described ones:

- **Service Scheduling Enabler (SSE):** in order to efficiently use the network resources, this enabler organizes the optimal schedule of MBMS service deliveries.
- **Content Management Enabler (CME):** works as an interface between the content and the service provider, controlling and defining policies for all the multicast/broadcast content.

8.6.2 Network Entities

Proxy – Call Session Control Function (P-CSCF)

Being the first contact point between the User Equipment and the IMS core, P-CSCF is responsible for handling the authorization needed by the end-user to allow it to access the desired service. This authorization is performed during the IMS session establishment, where the applications server or the S-CSCF check the user subscription information stored in the HSS after the response sent by PDF/PCRF to the MBMS authorization request initiated by the GGSN.

In case that the authorization is handled by the application server, it should act as follow:

- If the user is authorized to access the service, the application server replies with a 183 Session Progress or a 200 OK message including all the session related parameters.
- Otherwise, the application server answers with 403 Forbidden or 488 Not Acceptable Here message.

In case that the service is delivered over an MBMS bearer and the authorization is handled by the S-CSCF, it should acts as follows:

- The S-CSCF forwards the SIP response to the serving P-CSCF if the user is authorized
- Otherwise, the user gets a 403 forbidden message or 488 Not Acceptable Here sent by the S-CSCF

Interrogating – Call Session Control Function (I-CSCF)

This entity interacts with the HSS to obtain the name of the S-CSCF that serves the user equipment, thus reducing the delay of session establishment and the load of the S-CSCF.

Serving – Call Session Control Function (S-CSCF)

The S-CSCF, the main element of IMS core, provides session control for subscribers accessing services. It has the responsibility to interact with the HSS to get user information, such as authentication and profile. Based on this information, the S-CSCF forwards SIP messages to one or more Applications Servers or other final destinations.

Policy Detection Function / Policy and Charging Rules Function (PDF/PCRF)

Basically, the former MBMS Release 6 BM-SC membership function is now handled by the PDF/PCRF. Thus, policy decisions are made by this entity based on stored session parameters and media-related information obtained from the P-CSCF after receiving a bearer authorization request from the GGSN.

There are two ways how PDF/PCRF can provide authorization check for requested MBMS multicast service activations. On one hand, the Implicit Authorization consists in the PDF/PCRF processing the GGSN authorization request based on the session parameters and the service information acquired by the P-CSCF. On the other hand, in the Explicit Authorization, the authorization is processed with the PCRF making use of a new entity called Subscription Profile Repository (SRP) that stores all user allowed services.

The consequence of these distinct authorization processes is the splitting of the Gmb signalling into two interfaces. One of them is used between the GGSN and the MDFC responsible for MBMS bearer specific signalling and the other one is used for authorization check of multicast bearer activation requests, connecting the GGSN and the PDF/PCRF.

Media Delivery Function Controller (MDFC)

This entity is responsible for controlling the media processing functions and to provide the access of these functions to the application server. Furthermore, the MDFC acts as the BM-SC for proxy signalling between the GGSN.

8.7 IMS-MBMS Signalling

IMS was originally designed for unicast deliveries only among users. With the objective of supporting the use of broadcast and multicast bearers for multimedia content delivery to a group of IMS users, new interfaces were created and signalling procedures between entities were changed.

The next sessions present the most relevant signalling procedures between the IMS entities for the IMS-MBMS integrated architecture.

8.7.1 Service Activation

As in Release 6 MBMS, Service Activation is initiated when the user wants to be subscribed in a specific MBMS multicast service. In order to allow the network to configure the entire multicast path, this procedure is mandatory and needs to be initiated a long time before the start of the session.

The distinct processes that represent the IMS-MBMS Service Activation are shown in the next diagram:

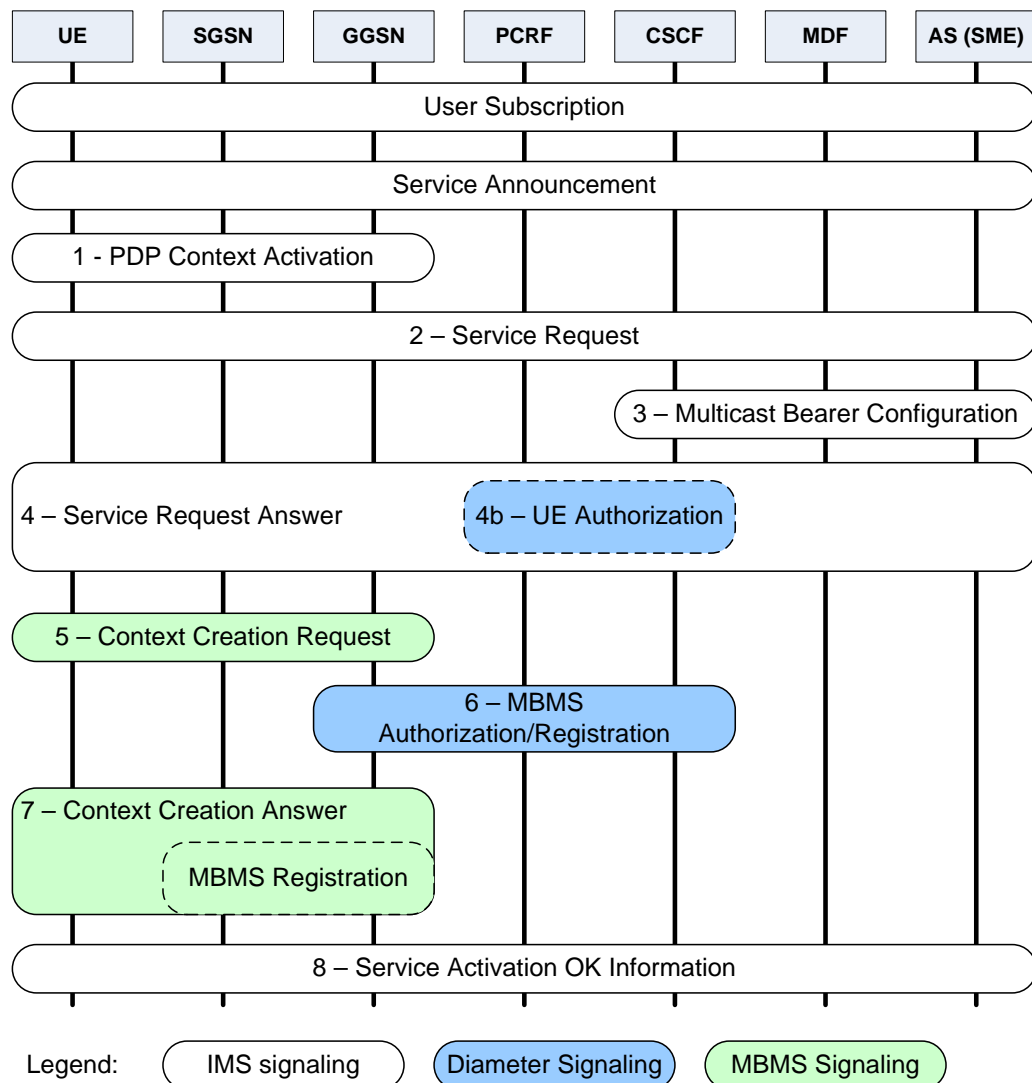


Figure 76 - Service Activation Signalling

Before the Service Activation procedure is initiated, User Subscription and Service Announcement processes must be concluded. Afterwards, if interested in accessing the announced service, the user starts the establishment of a PDP context (1) creating the bearer for sending the SIP message. This message, sent to the Application Server, announces the interest of the user for the specific service (2).

Based on the UE capabilities, the AS (SME) decides if the service will be transmitted in a multicast or unicast channel. If multicast, and in case of being the first UE to appeal for the service, the AS (SME) requests the configuration of the multicast bearer to the Media Delivery Function through CSCF (3). Then, the Media Delivery Function (MDF) answers with the Media Port Address and with the SDP to be used in the service. In a scenario that there are users already subscribed to the service, bearer configuration is not necessary.

In (4) the AS (SME) sends the service related information to the user. Meanwhile, and before reaching the UE, there is a dialog between the PCRF and the CSCF (4b) in order to authorize or deny the UE to access the desired service using the specific codecs. Then, it is the PCRF that forwards the authorization answer to the user.

After the user getting the response, the process of creating the UE context over the network is initiated (5). The end-user will create a MBMS UE context and sends parameters (such as IP multicast address, APN and MBMS bearer capabilities) allowing SGSN and GGSN to also create the UE context.

Following this, GGSN will seek the authorization decision for the activating user given by the PCRF (6), instead of BM-SC as in MBMS Release 6. Meanwhile, the GGSN registers itself in the MDF if the UE is the first one to show interest for that MBMS service through that specific GGSN.

After creating the UE context the GGSN replies to the SGSN (7) and, if not done before, a MBMS registration needs to be established between these two entities. Finally, the UE informs the AS (SME) the completion of the service activation (9).

8.7.2 Session Start

In IMS-MBMS integration, Session Start procedure triggered by the AS initiates when the MDF is ready to send data and to inform the network to prepare the network resources to transmit it. This procedure is mandatory since the network must first create the multicast path.

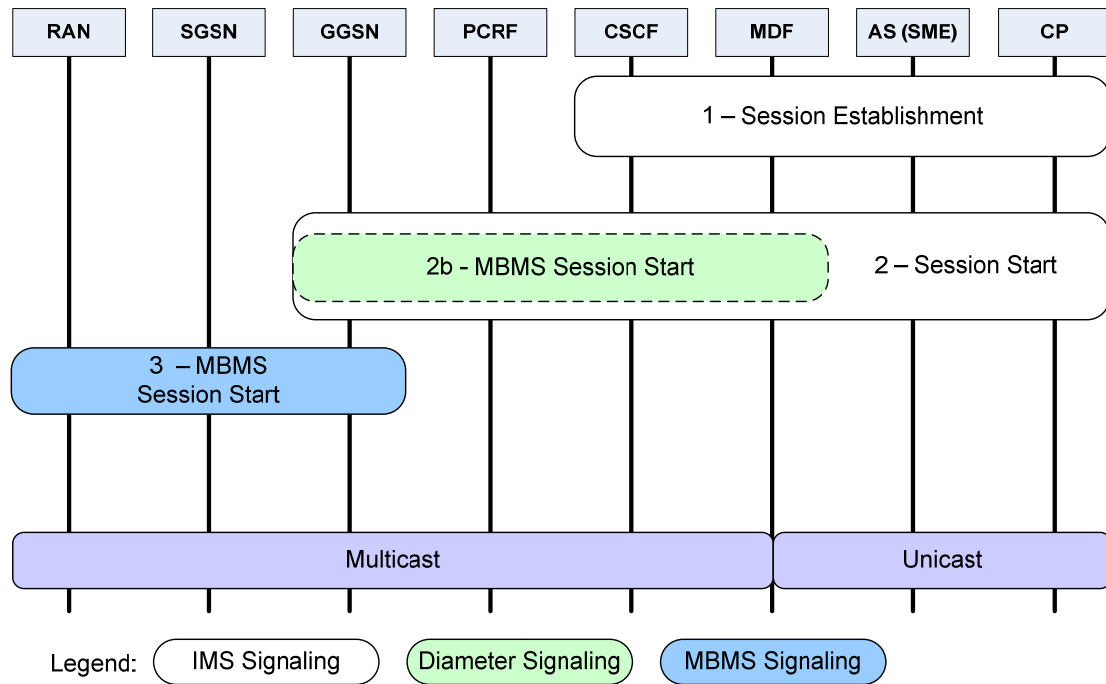


Figure 77 - Session Start Signalling

This procedure is initiated with the AS (SME) acting like a third party control between the Content Provider and the MDF. The AS (SME) sends a SIP Invite message to the CP and the CP replies with the Service Description Protocol (SDP), then forwarded to MDF (1). As a note, if the service is already being provided to other users, there is no need to initiate this process because the MDF was already triggered for this network.

Afterwards, Session Start phase (2) is initiated: the AS (SME) invites all the users subscribed to that service sending the SDP for the session, which includes the multicast address that is allocated to deliver the content with a specific quality. The MDF starts the MBMS Session Start procedure (2b), Packet Data Protocol (PDP) contexts are set to Active mode and network resources become reserved for the specified service and unavailable for other services. In this IMS-MBMS integration scenario, it is the Policy and Charging Rules Function (PCRF) that handles the service access authorization for the UE, instead of the BM-SC in MBMS Release 6.

Thereafter, and still in (2), the SIP signalling continues with the MDF sending a message containing the SDP Parameters. Then, the AS acknowledges both the MDF and the CP entities, thus resulting in the creation of the session between these two last entities.

Meanwhile, after receiving the session attributes from the GGSN, the SGSN saves them in the MBMS Bearer Context, set this context to the Active mode and forwards the session attributes to the linked RNC (3).

Having this procedure been completed, the service content from the CP should only be transferred to the UEs after a short period of time to assure that the establishment of the multicast channel is completed.

8.7.3 Session Stop

Having no more MBMS data to send, Session Stop (shown in the next figure) is triggered in order to release the multicast bearer and to free network resources.

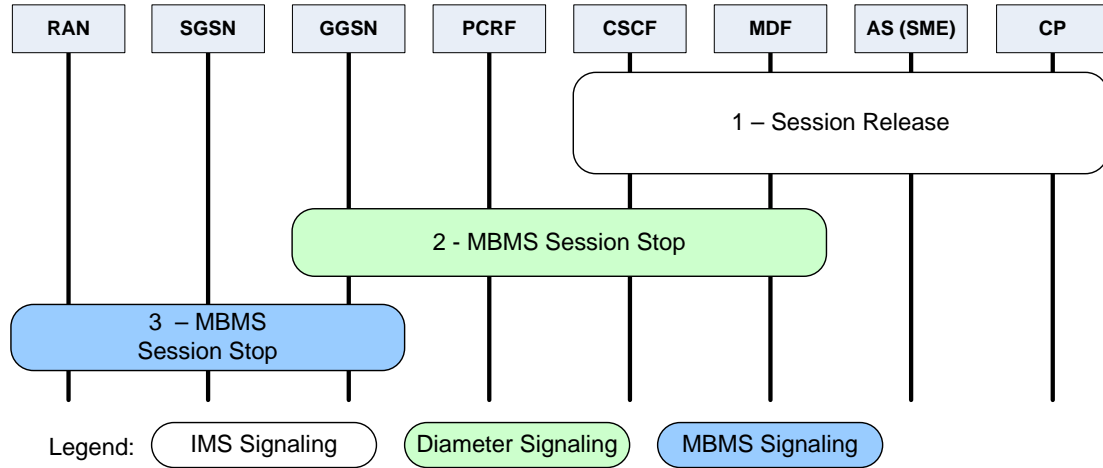


Figure 78 - Session Stop Signalling

As in Session Start procedure, the AS (SME) acts as a third party control between the MSF and the CP. So, it starts by sending messages to MDF and CP informing the release of the session between them (1). As a note, it is possible for the CP to trigger the Session Stop procedure by being the first to send the message and accordingly the AS release the session as described before.

In order to shutdown the multicast bearer, the MDF initiates the MBMS Session Stop procedure (2). Next, the GGSN informs the SGSN of the end of the transmission. The SGSN releases the bearer plane resources and sends a MBMS Session Stop message to the linked RNC, resulting in the releasing of the network resources (3).

8.7.4 Session Deactivation

When a user wants to leave a specified MBMS service, Service Deactivation procedure is initiated. This means that the user stops being a member of the multicast group subscribed to that service and therefore unable do get more content.

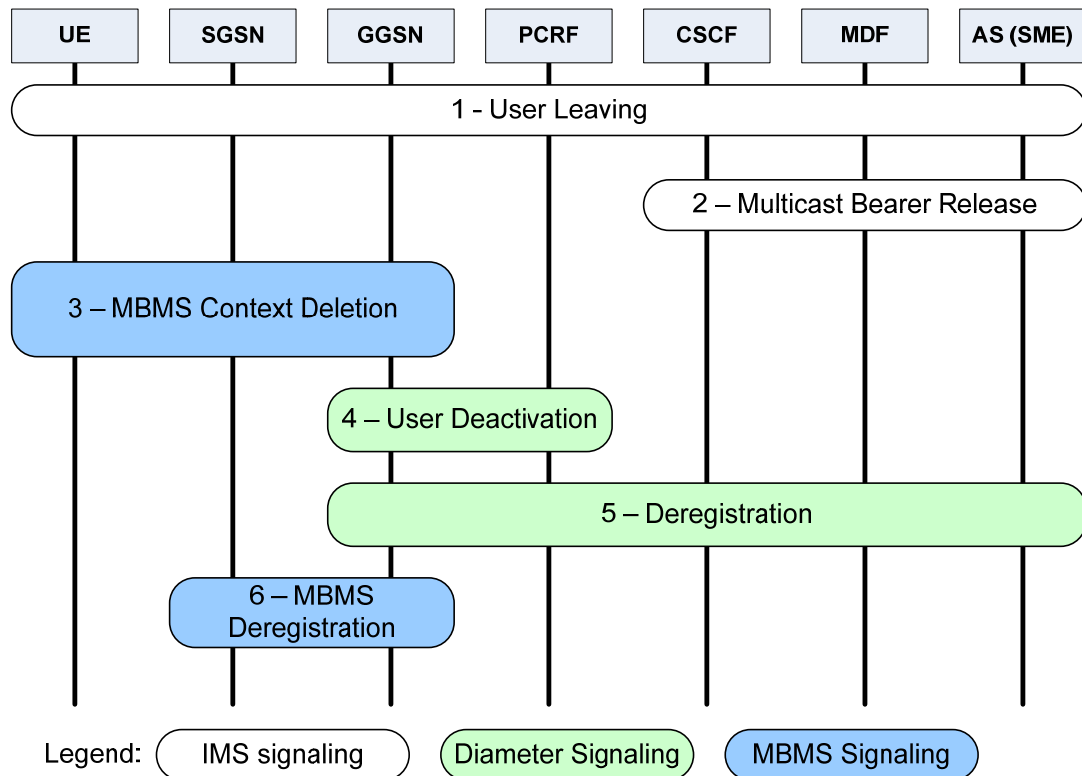


Figure 79 - Session Deactivation Signalling

With the objective of leaving the service, the UE sends a message to the AS (SME). At this point, the PCRF is aware on the UE intention and removes the user authorization (1). Meanwhile, if it was the last user subscribed to that service and therefore no reason to maintain the multicast bearer, the MDF will be instructed by the AS (SME) to release that specific bearer (2).

After receiving the leaving confirmation message from the CSCF, the user starts the MBMS Service Deactivation signalling in order to remove the established UE context, which will occur in the UE, in the SGSN and in the GGSN (3), resulting in the release of the network resources in the RAN. Next, after receiving the information about the deactivation from the GGSN, the PCRF removes all the data associated with the user (4).

In case of being the last user using the service, the GGSN must be deregistered from the MDF (5). Just before this deregistration procedure, the PCRF makes the resources available to other services.

In the end, if there are no more users linked to the SGSN, this entity proceeds with the MBMS Deregistration procedure.

9. Final Conclusions

Along the development of this whole work, a lot of new knowledge was acquired. Firstly, the basis of this work, i.e., mobile networks architecture, how they are structured, which entities are a part of them and how they work. Next, the whole evolution these networks have gone through in order to be able to support technologies such as the MBMS, related to third generation networks with all its inherent advantages and all the potentialities it can offer the user.

Being a leader as an AAA protocol for mobile systems, Diameter is widely used in technologies that involve authorization, authentication and accounting. Thus, it was obtained a strong notion of how this protocol works, of its great adaptation capabilities and above all its easy extensibility to other applications, as it was in this case the implementation of Gmb.

Through the inevitable reading of the specifications involving MBMS and the Gmb interface, notions of how new technologies that are appearing everyday and being developed are standardized and specified along with several discussions, changes and releases which involve groups like the 3GPP were obtained, until they become technologies mature and capable of being implemented.

Finally, the fact of being able to follow the evolution of a technology embodied in the next generation mobile networks is highly gratifying, concerning the achievement of new knowledge. Indeed, this technology is on the rise, it is capable of offering great capabilities and, most of all, it is extremely attractive.

Concerning the developed work in itself, it has gone through several phases until coming to its conclusion. For example, before getting to the final choice of the software platform to be used in its implementation, two others were studied. Having made that choice, an extensive study about the used platform architecture, how it works and how to extend it to other applications, as is the case of the Gmb, was carried out. Throughout the development of this work, the theory notions obtained both from the Diameter protocol and all the MBMS procedures got more solid.

In the end, following the conclusion of this implementation, the initial goals were entirely achieved. There is in fact the signalling performed by two machines which simulate communication carried out by the BM-SC and the GGSN entities, which compose the MBMS architecture. On that signalling, the Diameter protocol is involved as a means of control messages exchanging and, furthermore, the information included in the data structures in those entities, the MBMS Bearer Context and the MBMS User Context, is in fact added, altered and removed, according to the different procedures implemented on this work and specified in the Gmb signalling.

One can clearly state that the work developed will always be subject to extensions or further improvement, due to the ever-striving updates of the used software structure and even the new release of similar applications.

As stated before, attractive technologies like MBMS are always evolving. Like never before, there is the tendency of integrating different technologies in order to create a single system that gathers the full potentials of each one, resulting in a more flexible,

next-generation solutions that can easily cover real user needs. Therefore, the last focus of this work was on the current evolution of the MBMS, with the most relevant state-of-the-art approaches that involve the integration of MBMS with other technologies, such as MIMO Antennas, HSDPA, Single-Frequency Network, OFDMA and Dual Receiver.

The big emphasis though was on the IMS-MBMS integration proposed by C-MOBILE, where the capability of session control and capacity of dealing with multimedia content offered by the IMS technology combined with the broadcast/multicast content delivery of MBMS can result in a global converged network architecture that can offer modern interactive services for massive distribution. Moreover, the main purpose of focusing on this IMS-MBMS integrating approach was to analyze its signalling processes between the ones in MBMS Release 6 that were studied and implemented in the first stage of this work.

References

- [1] 3GPP TS 29.061 V6.9.0, Interworking between the Public Land Mobile Network (PLMN) supporting packet based services and Packet Data Networks (PDN) (Release 6)
- [2] 3GPP TS 23.246 v6.10.0, Multimedia Broadcast/Multicast Service (MBMS), Architecture and functional description (Release 6)
- [3] RFC 3588 - Diameter Base Protocol
- [4] OpenBloX™ - User Manual, Traffix Systems – www.traffixsystems.com
- [5] WCDMA for UMTS, Harri Holma and Antti Toskala, John Wiley & Sons Ltd, 2004
- [6] MBMS White Paper, TeliaSonera, MediaLab, 2004
- [7] Introduction to Diameter, Interlink Networks, 2002
- [8] Internet Assigned Number Authority - www.iana.org
- [9] MBMS Dual Receiver - <http://www.wipo.int/pctdb/en/wo.jsp?wo=2007091823>
- [10] High-Speed Downlink Packet Access - Wikipedia
<http://en.wikipedia.org/wiki/HSPDA>
- [11] MIMO Spatial Multiplexing For E-MBS
http://www.wirelessman.org/tgm/contrib/C80216m-08_124r1.pdf
- [12] Single Frequency Network – Wikipedia
http://en.wikipedia.org/wiki/Single-frequency_network
- [13] OFDMA – Runcom solutions - www.runcom.com
- [14] Orthogonal Frequency-Division Multiple Access: is it the multiple access system of the future? Srikanth S, Kumaran V., Manikandan C. - <http://comm.au-kbc.org>
- [15] OFDMA – Wikipedia
http://en.wikipedia.org/wiki/Orthogonal_frequency_division_multiple_access
- [16] Introduction to IP Multimedia Subsystem (IMS), IBM -
<http://www.ibm.com/developerworks/webservices/library/ws-soa-ipmultisub1/>
- [17] C-MOBILE Project web page - <http://c-mobile.ptinovacao.pt>
- [18] C-MOBILE, Deliverable 4.1, "Detailed research objectives and requirements for the core network enhancements and new entities", 2006
- [19] C-MOBILE, Deliverable 4.2, "Draft Specification & Requirements of Enhanced Core Network Elements", 2007
- [20] C-MOBILE, Deliverable 4.3, "Specification of Enhanced Core Network Elements", 2007

- [21] Enabling IMS with Multicast and Broadcast Capabilities, Adel Al-Hezmi, Michael Knappmeyer, Björn Ricks, Filipe Cabral Pinto and Ralf Tönjes, 2008
- [22] Mobile TV Workshop presentation, Filipe Cabral Pinto – PT Inovação, April 2008
- [23] 3GPP – 3rd Generation Partnership Project - <http://www.3gpp.org>
- [24] RADIUS - Remote Authentication Dial In User Service (RADIUS) standard - <http://tools.ietf.org/html/rfc2865>
- [25] DVB-H Website and specification, <http://www.dvb-h.org/technology.htm>
- [26] MediaFLO – Wikipedia - <http://en.wikipedia.org/wiki/MediaFLO>
- [27] MediaFLO technology description - <http://www.mediaflo.com/mediaflo/index.html>
- [28] Digital Multimedia Broadcast (DMB) technology description – <http://www.worlddab.org/technology>