







## **DEDICATION**

I dedicate this work to God for saving my life and His Blessings bestowed upon me to complete this project. I also dedicate this work to my parents (Mbazima and Zondeka), my brothers, my sisters and my best friend, Maropeng Ngobeni.

## ACKNOWLEDGEMENTS

I express my sincere appreciation and gratitude to the following people and institutions, for their contribution to this project:

-  To TUT for offering me an opportunity to further my studies. Most importantly I would like to thank Prof Ben van Wyk (in the Dean's office) and the financial bureau for their financial support.
-  To FSATIE for their support academically and financially and my supervisors Damien CHATELAIN, James OYEDAPO and Anish KURIEN.
-  To the National Research Foundation (NRF) for financial support.
-  To Prof Zeglache DJAMAL and Dr Marc GIROT-GENET from the Institute National des Télécommunications, for their hospitality and support during my research period in France and also to fellow INT postgraduate students.
-  To Région Ile-de-France for offering me the financial support to further my studies in France.
-  To Ecole Supérieure d'Ingénieurs en Electronique et Electrotechnique (ESIEE) – Paris as well as Mr Pierre ABEILLÉ and Fabiane DELLA ZUANA for their invaluable support during my studies and stay in France.
- To fellow FSATIE and TUT students.

## ACRONYMS

<b>3GPP:</b>	Third Generation Partnership Project
<b>API:</b>	Application Programming Interface
<b>AS:</b>	Application Server
<b>CDC:</b>	Connected Device Configuration
<b>CDR:</b>	Charging Data Record
<b>CLDC:</b>	Connected Limited Device Configuration
<b>CSCF:</b>	Call State Control Function
<b>DHCP:</b>	Dynamic Host Configuration Protocol
<b>DNS:</b>	Domain Name System
<b>GGSN:</b>	Gateway GPRS Support Node
<b>GPRS:</b>	General Packet Radio Service
<b>HSS:</b>	Home Subscriber Server
<b>IBCF:</b>	Interconnect Border Control Function
<b>I-CSCF:</b>	Interrogating-CSCF
<b>IETF:</b>	Internet Engineering Task Force
<b>IMS:</b>	Internet Protocol Multimedia Subsystem
<b>INT:</b>	Institute National des Télécommunications
<b>ISIM:</b>	IP multimedia Services Identity Module
<b>J2ME:</b>	Java 2 Micro-Edition
<b>JCP:</b>	Java Community Process
<b>JSR:</b>	Java Specification Request
<b>MAGNET:</b>	My personal Adaptive Global Network
<b>MGCF:</b>	Media Gateway Control Function
<b>MIDP:</b>	Mobile Information Device Profile
<b>MRFC:</b>	Multimedia Resource Function Controller
<b>MSMP:</b>	MAGNET Service Management Platform
<b>NGN:</b>	Next Generation Network
<b>P-CSCF:</b>	Proxy-CSCF

<b>PDP:</b>	Packet Data Protocol
<b>PSTN:</b>	Public Switching Telecommunications Network
<b>QoS:</b>	Quality of Service
<b>RFC:</b>	Request for Comments
<b>S-CSCF:</b>	Serving-CSCF
<b>SDP:</b>	Service Description Protocol
<b>SIP:</b>	Session Initiation Protocol
<b>THIG:</b>	Topology Hiding Inter-network Gateway
<b>TISPAN:</b>	Telecom and Internet Services and Protocols for Advanced Networks
<b>UA:</b>	User Agent
<b>UE:</b>	User Equipment
<b>UICC:</b>	Universal Integrated Circuit Card
<b>UMTS:</b>	Universal Mobile Telecommunications System
<b>UPnP:</b>	Universal Plug and Play
<b>URI:</b>	Uniform Resource Identifier
<b>USIM:</b>	Universal Subscriber Identifier Module

## LIST OF FIGURES

	<b>PAGE</b>
Figure 2.1: IMS relationship with other communication systems.....	9
Figure 2.2: The IMS Architecture (Radvision, 2006) .....	11
Figure 2.3: The 3GPP IMS Architectural Elements.....	12
Figure 2.4: UE Initiated PDP Context Activation (Pouse <i>et al</i> , 2003).....	16
Figure 2.5: P-CSCF Discovery using DHCP Procedure (3GPP TS 23.228, 2006).....	17
Figure 2.6: INVITE routing (adapted from Poikselkä <i>et al</i> , 2004) .....	18
Figure 2.7: Release 6 user identities (Camarillo & García-Martín, 2006:43) .....	19
Figure 2.8: TISPAN IMS Functional Elements (Newport-Networks, 2005).....	21
Figure 2.9: The Internet multimedia protocol stack (Johnston, 2001:4) .....	22
Figure 2.10: SIP Trapezoid (Poikselkä <i>et al</i> , 2006) .....	23
Figure 2.11: SIP message format (Adapted from: Rosenberg <i>et al</i> , 2002) .....	24
Figure 2.12: Personal Network (MAGNET, 2006).....	27
Figure 2.13: Shipments of Naked SIP versus Closed IMS handsets (Bublely, 2006).....	29
Figure 2.14: FOKUS OpenIC Lite (FOKUS, 2007).....	30
Figure 2.15: IMS-Communicator Screenshot (PT Inovação, 2007).....	31
Figure 2.16: UCT IMS Client Screenshot (Waiting & Good, 2007).....	32
Figure 3.1: Sun Java WTK 2.5 API Selection Window .....	36
Figure 3.2: J2ME Stack (Li & Knudsen, 2005: 2).....	39
Figure 3.3: IMS and the applicability of the SIP specifications (O'Doherty, 2003) .....	40
Figure 3.4: Simplified class diagram of SIP API for J2ME (JCP JSR 180, 2004).....	41
Figure 3.5: Sending SIP request (JCP JSR 180, 2004).....	43
Figure 3.6: Receiving a SIP request (JCP JSR 180, 2004).....	44
Figure 4.1: IMS Client Stack (Based on Li & Knudsen, 2005:2) .....	45
Figure 4.2: IMS Client position in relation to IMS CN.....	46
Figure 4.3: Session Establishment Main Flow diagram.....	47
Figure 4.4: IMS Client Sequence Diagram (3GPP TR 24.930, 2007) .....	48
Figure 4.5: IMS Client Display before Registration.....	50
Figure 4.6: IMS Client Busy Display .....	50
Figure 4.7: IMS Client Display after Registration .....	51
Figure 4.8: IMS Client Invite Display.....	52
Figure 4.9: An IMS Client Chatting Display.....	55
Figure 4.10: INT IMS Testbed .....	56
Figure 4.11: Sun Wireless Toolkit – KToolbar .....	58
Figure 5.1: Experimental Setup at INT IMS Testbed.....	61
Figure 5.2: INT IMS Testing Environment.....	64
Figure 5.3: Demonstration of How to Establish a Session.....	65
Figure 5.4: REGISTER Request Trace .....	66
Figure 5.5: INVITE Request Trace .....	68

Figure 5. 6: 100-Trying Response Trace .....	69
Figure 5.7: 183-Session Progress Response Trace .....	70
Figure 5.8: PRACK Request Trace .....	70
Figure 5.9: 200-OK Response for PRACK Trace .....	71
Figure 5.10: UPDATE Request Trace.....	72
Figure 5.11: 200-OK Response for UPDATE Trace.....	72
Figure 5.12: ACK Request Trace .....	73
Figure 5.13: MESSAGE Request Trace .....	74

## LIST OF TABLES

	<b>PAGE</b>
Table 2. 1: Six RFC 3261 SIP Request Methods .....	25
Table 2. 2: Other SIP Request Method Extensions .....	25
Table 3.1: Mobile Development Platforms (Mao, Talevski & Chang, 2007) .....	35
Table 4.1: INVITE Request Trace.....	52
Table 5.1: Hardware Specifications .....	61
Table 5.2: Software Specifications .....	62
Table 5.3: INT IMS Testbed Specifications .....	63
Table 5.4: SDP Attributes.....	67
Table 5.5: Comparison of Light-Weight and Open IMS Client Functionalities .....	75

# CONTENTS

	PAGE
DEDICATION .....	i
ACKNOWLEDGEMENTS .....	ii
ACRONYMS .....	iii
LIST OF FIGURES .....	v
LIST OF TABLES .....	vii
CONTENTS .....	viii
CHAPTER 1 .....	1
INTRODUCTION .....	1
1.1 Project Background .....	1
1.2 Problem Statement.....	2
1.2.1 Sub-Problem 1 .....	2
1.2.2 Sub-Problem 2 .....	2
1.2.3 Sub-Problem 3 .....	3
1.3 Delimitations .....	3
1.4 Research Methodology .....	3
1.5 Benefits of the Study .....	4
1.6 Brief Overview of this Dissertation.....	4
CHAPTER 2 .....	5
LITERATURE SURVEY.....	5
2.1 INTRODUCTION .....	5
2.2 THE IMS AND SIP STANDARDIZATION BODIES .....	5
2.2.1 The 3 <sup>rd</sup> Generation Partnership Project (3GPP) .....	5
2.2.2 European Telecommunications Standard Institute (ETSI).....	6
2.2.3 International Telecommunications Union (ITU).....	6
2.2.4 The Internet Engineering Task Force (IETF).....	6
2.2.5 Java Community Process (JCP).....	7
2.3 THE IP MULTIMEDIA SUBSYSTEM – IMS .....	8
2.3.1 IMS Background and History.....	8
2.3.2 IMS with other Communication Systems.....	9
2.3.3 The IMS Architecture .....	10
2.3.4 The IMS Network Elements .....	11
2.3.5 The IMS Concept and Procedure.....	15
2.3.6 User Identification in IMS.....	18
2.3.7 IMS in the Next Generation Networks.....	20
2.4 SESSION INITIATION PROTOCOL – SIP .....	21
2.4.1 SIP Background and History .....	21
2.4.2 SIP Architecture .....	22
2.4.3 SIP Messages Format .....	23
2.4.4 SIP Request Methods .....	25
2.4.5 SIP P-Headers for IMS .....	26
2.5 PERSONAL NETWORKS – PN .....	26



2.6	INTRODUCING THE MAGNET BEYOND PROJECT .....	28
2.7	RELATED WORK.....	28
2.7.1	FOKUS Open IMS Client Lite .....	29
2.7.2	IMS-Communicator.....	30
2.7.3	UCT IMS Client .....	31
2.8	LIGHT-WEIGHT IMS CLIENT CONCEPT.....	32
2.9	CONCLUSION .....	33
CHAPTER 3.....		34
TOOLS FOR DEVELOPING JAVA BASED PORTABLE APPLICATIONS.....		34
3.1	INTRODUCTION .....	34
3.2	DEVELOPMENT TOOLS.....	34
3.2.1	Java Development Kit (JDK) .....	34
3.2.2	J2ME Sun Wireless Toolkit.....	35
3.2.3	EclipseME .....	36
3.3	JAVA 2 MICRO-EDITION PLATFORM.....	37
3.3.1	Configurations .....	37
3.3.2	Profiles.....	38
3.4	JAVA LIBRARIES FOR SIP PROGRAMMING .....	40
3.4.1	SIP API for J2ME.....	40
3.5	CONCLUSION .....	44
CHAPTER 4.....		45
SYSTEM DEVELOPMENT AND IMPLEMENTATION .....		45
4.1	INTRODUCTION .....	45
4.2	SYSTEM DEVELOPMENT.....	45
4.2.1	SIP Registration.....	49
4.2.2	Session Establishment .....	51
4.2.3	Session Description Protocol (SDP) Offer/Answer.....	53
4.2.4	Communication Mode .....	54
4.2.5	Cancelling an Early Session .....	55
4.2.6	Ending Session .....	55
4.3	SYSTEM IMPLEMENTATION.....	56
4.3.1	IMS Network Testbed .....	56
4.3.2	System Simulation.....	57
4.4	CONCLUSION .....	59
CHAPTER 5.....		60
SYSTEM EXPERIMENTS AND RESULTS .....		60
5.1	INTRODUCTION .....	60
5.2	EXPERIMENTAL SETUP .....	60
5.3	TESTING REQUIREMENTS.....	61
5.3.1	Hardware Requirements .....	61
5.3.2	Software Requirements.....	62
5.3.3	INT IMS Testbed Specifications .....	63
5.4	EXPERIMENTAL DEMONSTRATION .....	63
5.5	RESULTS DISCUSSION .....	65
5.5.1	Registration Request.....	66
5.5.2	Invite Request .....	67
5.5.3	Trying Response .....	68

5.5.4	Session Progress Response .....	69
5.5.5	PRACK Request .....	70
5.5.6	UPDATE Request .....	71
5.5.7	ACK Request .....	72
5.5.8	Communication Mode .....	73
5.6	LIGHT-WEIGHT AND OPEN IMS CLIENT IMPLEMENTATIONS.....	74
5.7	CONCLUSION .....	75
CHAPTER 6.....		77
CONCLUSION AND FUTURE WORK.....		77
6.1	SUMMARY OF FINDINGS.....	77
6.2	DISCUSSION OF PROBLEMS .....	77
6.3	FINDINGS AND CONCLUSIONS.....	78
6.4	SUMMARY OF CONTRIBUTIONS .....	78
6.5	SUGGESTIONS FOR FUTURE WORK .....	79
REFERENCES .....		80
APPENDIX A .....		85
SIP MESSAGE FLOW .....		85
A.1:	SIP Registration Message Flow.....	85
A.2:	Session Establishment Message Flow .....	85
A.3:	Session Termination Message Flow .....	87
APPENDIX B.....		88
JAVA CODE TO CREATE A DISPLAY OF J2ME.....		88
APPENDIX C.....		89
JAVA CODE .....		89

## CHAPTER 1

### INTRODUCTION

#### 1.1 Project Background

In recent years Internet usage and services, accessed through both fixed and mobile networks, have experienced rapid growth. This growth was driven by the capability of the Internet to provide many new services seamlessly to the users at any time. These Internet services continue to grow because the open protocols used are available on the web to any service developer (Camarillo & García-Martín, 2006:5). It is estimated that there are over one billion cellular users worldwide (Castro, 2004:1). As the number of cellular users continues to grow, more efforts are made to draw the cellular technology closer to the all-IP telecommunication technologies that offer Quality of Service (QoS) to users, and appropriate charging schemes for multimedia services to network operators. With Internet Protocol (IP) becoming spreading through in the backbone, the challenge of integrating voice and data services in the fixed and mobile environments becomes more formidable (Castro, 2004:5).

The Internet Protocol Multimedia Subsystem (IMS) is a global, access-independent and standard-based IP connectivity and service control architecture that enables various types of multimedia services to end-users using common Internet-based protocols (Poikselkä *et al*, 2006:4). The IMS was introduced by the Third Generation Partnership Project (3GPP) standardization body for the 3G networks in Release 5 Technical Specification (TS) (3GPP TS 23.228, 2006). Since its introduction, IMS has been adopted by other major telecommunication standardization bodies in mobile and fixed networks as the basis for the Next Generation Network (NGN). The Session Initiation Protocol (SIP) (Rosenberg *et al*, 2002) was selected as the main standardized signalling protocol for the IMS. Unlike traditional IP-based networks, the IMS guarantees end-to-end QoS in the network and creates an infrastructure that enables the fast deployment of new IP-based services and flexible billing, while still maintaining compatibility with existing applications (Fabini *et al*, 2006).

This project seeks to develop the IMS Client that is compliant with the 3GPP, European Technically Standard Institute (ETSI) Telecom and Internet Services and Protocols for Advanced Networks (TISPAN) and the Internet Engineering Task Force (IETF) recommendations and specifications. The developed IMS Client was intended to be integrated to the service discovery and service control execution environment of Personal Networks (PN), which will be contributing towards the My personal Adaptive Global Network (MAGNET) Beyond project (which will be discussed later).

## **1.2 Problem Statement**

The main objectives of this project are as follows:

- To improve, qualify and finalize the implementation of the architecture of service discovery and management platform for Personal Networks (PN) and the federations of Personal Networks.
  
- To analyze the global service discovery architecture and then design an IMS Client compliant with the 3GPP, ETSI-TISPAN and IETF as the major IMS standardization bodies.

The following sub-problems that were identified formed the basis for the study:

### **1.2.1 Sub-Problem 1**

To study the concept, architecture and procedure of the IMS and SIP systems as specified and recommended by the 3GPP, ETSI-TISPAN and IETF.

### **1.2.2 Sub-Problem 2**

To study the available open source SIP architecture, entities and networks with the aim of using the open source SIP User Agents (UA) as a prototype for building the proposed IMS Client.

### **1.2.3 Sub-Problem 3**

To develop an IMS Client that is compliant with 3GPP and ETSI-TISPAN recommendations and specifications and also compliant with SIP IETF RFC 3261 and other SIP extension Request For Comments (RFCs).

## **1.3 Delimitations**

- 1.3.1 This study will not deal with IMS user authentication during the registration stage. This is due to the current setup of the IMS testbed platform that was used for testing the IMS Client.
- 1.3.2 This study assumes the existence of IP security between the IMS Client and the IMS core network.
- 1.3.3 This study does not take into account the use of the IP multimedia Services Identity Module (ISIM).

## **1.4 Research Methodology**

In order to achieve the objectives of this study, the following steps were followed:

- An in-depth literature study of the 3GPP IMS by reading the Technical Specifications, books and published work, related to the IMS was conducted. The same literature study was conducted for the IETF SIP.
- Java Specification Request (JSR) number 180 (JSR 180) was identified and studied as the core SIP API to be used for the development of the IMS Client.
- Sun Wireless Toolkit was used as the Java 2 Micro-Edition (J2ME) development platform for the client. EclipseME Integrated Development Environment (IDE) was used to edit the Java code.
- Testing and benchmarking of the developed lightweight of the IMS client within the INT IMS Testbed
- The experiments were carried out to evaluate the performance of the IMS Client within the Local Area Network (LAN) and WLAN at the INT laboratory
- And the final step was the discussion of the results

## 1.5 Benefits of the Study

The contribution of this work is that an innovative lightweight IMS Client of which the architecture and specifications are based on the 3GPP, IETF and ETSI-TISPAN standardization bodies has been developed. This IMS Client is targeted at power and memory-limited devices and can easily be deployed within mobile devices. As a result, this work will contribute to the standardization of service discovery in Personal Networks, which form part of the MAGNET Beyond project.

## 1.6 Brief Overview of this Dissertation

This dissertation is structured as follows:

*Chapter 2:* Presents the literature survey of the IMS and SIP. The chapter also introduces the Personal Network and MAGNET Beyond project. The IMS that was introduced by the 3GPP has now been adopted by most standardizing bodies, while SIP was chosen by the 3GPP as the core signalling protocol for IMS. Related work in the development of IMS Client is also presented in this chapter.

*Chapter 3:* Presents the literature survey of Java Micro-Edition (J2ME) development platform. J2ME is the second revolution of Java platform used for the development of small devices such as the Television set-top boxes and mobile phones.

*Chapter 4:* Provides the IMS Client design and development. The IMS Client was developed under the J2ME platform using Java code.

*Chapter 5:* Outlines the experimental setup and requirements. The chapter also presents the system experiments, evaluation and discussion of the results.

*Chapter 6:* Concludes the whole dissertation, and contains suggestions for future work.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 INTRODUCTION**

This chapter reviews the literature related to IMS, SIP and Personal Network (PN). The idea is to familiarise the reader with the major standardization bodies responsible for the telecommunication and Internet technologies, the architecture, network elements, the concepts and services that are enabled by both the IMS and SIP. The architecture, network elements and procedures of IMS and SIP are discussed in this chapter. The chapter also introduces MAGNET Beyond project.

#### **2.2 THE IMS AND SIP STANDARDIZATION BODIES**

There are several internationally recognized telecommunication and Internet standardization bodies responsible for the standardization of the IMS, SIP and Java technologies. This section cites major IMS, SIP and Java standardization bodies.

##### **2.2.1 The 3<sup>rd</sup> Generation Partnership Project (3GPP)**

According to the partnership agreement signed in December 1998, the 3GPP was formed in 1998 by the standardization bodies from Europe, Japan, USA and China to establish a 3<sup>rd</sup> generation (3G) mobile system based on evolved GSM core networks and the radio access technologies. The main purpose of 3GPP is to prepare, approve and maintain globally applicable Technical Specifications (TS) and Technical Reports (TR) for 3G mobile systems. IMS was introduced in Release 5 (3GPP TS 23.228, 2006) as part of 3GPP UMTS specification, and was mainly dedicated for 3G mobile systems. Release 6 and beyond of 3GPP specification added wireline support to IMS architecture.

### **2.2.2 European Telecommunications Standard Institute (ETSI)**

ETSI has established the Telecoms and Internet converged for Services and Protocols for Advanced Network (TISPAN) project as their core competence centre for the Next Generation Networks (NGN). TISPAN facilitates the smooth migration of fixed (wireline) networks from Circuit-Switched (CS) networks to Packet-Switched (PS) networks with an architecture that can serve in both to create the NGN. TISPAN has adopted the IMS architecture given in the 3GPP Release 6 but has added wireline access to the IMS (ETSI TR 180.001, 2006). Essentially, TISPAN adopts the 3GPP IMS architecture standard for SIP-based applications, but has added further functional blocks and subsystems to handle non-SIP applications and other requirements of fixed networks not addressed by IMS (Hills, 2006).

### **2.2.3 International Telecommunications Union (ITU)**

ITU-T created an NGN Focus Group (NGN-FG) in May 2004 to work on the specifications of NGN for fixed line access based on 3GPP IMS. The process of fitting IMS into ITU's NGN is performed by the ETSI TISPAN group, which is in turn responsible for all aspects of standardization for present and future converged networks (Fabini *et al*, 2006).

### **2.2.4 The Internet Engineering Task Force (IETF)**

The IETF is an open forum consisting of individuals working with vendors, operators, researchers and other interested individuals who make technical and other contributions to the engineering and evolution of the Internet and its technologies (Soliman, 2004:xx). There are two documents used within the IETF, namely the Internet-Drafts (I-Ds) and the Request for Comments (RFC). A standard that begins life as an I-D, progresses to an RFC once there is consensus and there are working implementations of the protocol. When changes are made in a protocol or new versions come out, a new RFC document with a new number is issued, which renders the old RFC obsolete (Johnston, 2001:2).

The 3GPP have established collaboration with the IETF to make sure that the protocols developed for IMS meet their requirements. This collaboration has been documented in RFC 3113. The IETF is responsible for the standardization of SIP, RFC 3261 being the core SIP



specification. Several RFCs were developed as the extension to the core RFC 3261 in order to meet 3GPP IMS requirements. The selection of SIP as the core IMS signalling protocol makes the IMS network to be regarded as an advanced SIP network (Safavian, 2006).

### **2.2.5 Java Community Process (JCP)**

Specifications for Java platforms are developed under the aegis of the JCP. A specification starts off as a Java Specification Request (JSR). An expert group consisting of representatives from interested companies is formed to create the specification. The JSR then passes through various stages in the JCP before it is finalised. Every JSR is assigned a number. The JCP is also involved in the standardization of Java APIs to facilitate the easy and fast deployment of IP services using SIP.

The following are JSRs defined by JCP for SIP:

- **JAIN SIP:** JSR 32 determines the Java Advanced Intelligent Network (JAIN) SIP specification. Sun Microsystems is the specification lead in conjunction with other expert companies and individuals. JAIN SIP can be implemented on Personal Digital Assistance (PDA), SIP phones and desktops.
- **SIP Servlet:** JSR 116 defines the SIP Servlet specification. Dynamicsoft is the specification lead in conjunction with other expert companies. SIP Servlet is implemented in web tier enterprise application servers, which has the benefit of converged SIP and HTTP applications (O'Doherty, 2003).
- **SIP for J2ME:** JSR 180 determines the SIP for J2ME specification - SIP interface for small platforms. Nokia Corporation is the specification lead in conjunction with other expert companies and individuals. SIP's acceptance as the protocol of choice by the IMS architecture within the 3GPP, highlights the value of mobile devices understanding and communicating SIP (O'Doherty, 2003). SIP for J2ME API can be incorporated into mobile handsets.

## **2.3 THE IP MULTIMEDIA SUBSYSTEM – IMS**

IMS is a global, access-independent and standard-based IP connectivity and service control architecture that enables various types of multimedia services to end-users using common Internet-based protocols (Poikselkä *et al*, 2006:4). IMS architecture is at the heart of the convergence of voice, data, fixed and mobile networks and is based on a wide range of IETF protocols, such as SIP. IMS combines and enhances these protocols to allow real-time services in addition to 3GPP mobile Packet-Switched (PS) domain and the wireline NGN.

The main vision of IMS is to bring the Fixed Mobile Convergence (FMC) to reality by enabling the provision of multimedia services to both fixed and mobile subscribers (3G Americas, 2007). This vision has been achieved by bridging the gap that exists between the traditional telecommunications provided by both cellular and Public Switched Telecommunication Network (PSTN) operators and the Internet. IMS provides the key functionalities required to enable new IP services via mobile networks, taking into account the complexity of managing mobility, multimedia, constraints of the underlying network and the multitude of emerging applications (3G Americas, 2004). With IMS, competition between fixed and mobile users will exist because all the service providers and operators will be targeting a single customer to offer similar services.

In this section the background and history of IMS are discussed, followed by the position of IMS in relation to the existing communication systems. Also covered in this section is IMS architecture, core network elements and IMS enabled services.

### **2.3.1 IMS Background and History**

After having released their first specification in 1999 (3GPP R99), the 3GPP started to specifying Release 2000 which included the All-IP systems that was later renamed the IMS. After realizing that the development of IMS could not be completed during the year 2000, the agreement was reached to split Release 2000 into Release 4 and Release 5 specifications (Poikselkä *et al*, 2004:6). After Release 4 has been frozen and completed, IMS was introduced by the 3GPP within Release 5 in the year 2002.

The IMS specified in Release 5 was only based on the 3G mobile systems, and only compatible to IP version 6 (IPv6). The 3GPP continued their IMS specifications and enhancements in Releases 6 and 7 (3GPP TR 24.930, 2007) of their specifications. Release 6 IMS (3GPP TS 23.228, 2007) included the interworking with the CS networks and other IP based networks. It also introduced more service provisioning which included Presence service, Push to-talk-over Cellular (PoC) and Instant Messaging (IM) servers. Release 7 specification that was released in 2006 included further IMS enhancements.

### 2.3.2 IMS with other Communication Systems

IMS is positioned above different existing communication systems, as depicted in Figure 2.1 below, hence it is regarded as an excellent vehicle for FMC (Ericsson, 2004).

Cellular communication technologies have evolved so rapidly that today they are able to offer IP connectivity to cellular users everywhere. With IP connectivity capabilities, network operators can enable their customers to access IMS services irrespective of the type of terminal used.

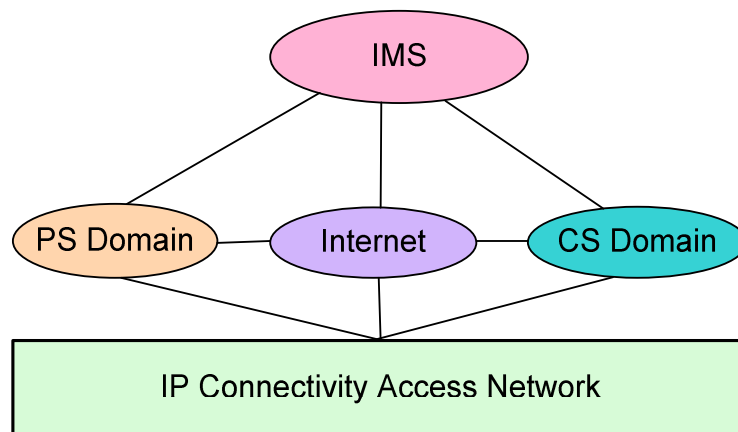


Figure 2.1: IMS relationship with other communication systems

Accessibility to IMS services is not limited to users with IP connectivity terminals, but also to CS network users. IMS introduces multimedia session control in the PS domain and the CS domain (Poikselkä *et al*, 2004). The IMS technology promises added advantages to all the

stakeholders in the NGNs, those being network operators, value-added service providers, enterprises and customers.

There are four additional key functionalities that mark IMS as the future technology in a comprehensive service and application-oriented network (Magedanz, Witaszek & Knuettel, 2005):

- IMS provides easy and efficient ways to integrate different services, even from third parties.
- IMS enables the seamless integration of legacy services, therefore is designed for consistent interactions with CS domains.
- IMS supports a mechanism to negotiate QoS. Within a session you the user can ask for QoS for certain Packet Data Protocol (PDP) Contexts on the critical 3G air interface.
- IMS provides appropriate charging mechanisms.

### **2.3.3 The IMS Architecture**

The IMS architecture comprises four planes as depicted in Figure 2.2 below. The device plane consists of the user terminals used to access the IMS services. An IMS-capable device or an IMS Client can be used to access IMS network. The device plane is the focus of this project. The device plane includes smart-phones, switch-phones and other advanced IP phones. The standardization of this plane also forms part of the 3GPP and other major IMS standardization bodies, for both wireless and wireline networks.

The transport plane refers to the access network used by IMS terminals to access the IMS network. Both users in the wireless and wireline network can access the IMS services. The Call Session Control Function (CSCF) forms the core of the IMS control layer. The CSCF is made up of the Proxy-CSCF (P-CSCF), Interrogating-CSCF (I-CSCF) and the Server-CSCF (S-CSCF). The Home Subscriber Subsystem (HSS) database is another element within the control layer. Other elements forming the IMS control plane includes the Breaking Gateway

Control Function (BGCF), Media Resource Function (MRF) and others, which will be discussed in the next section.

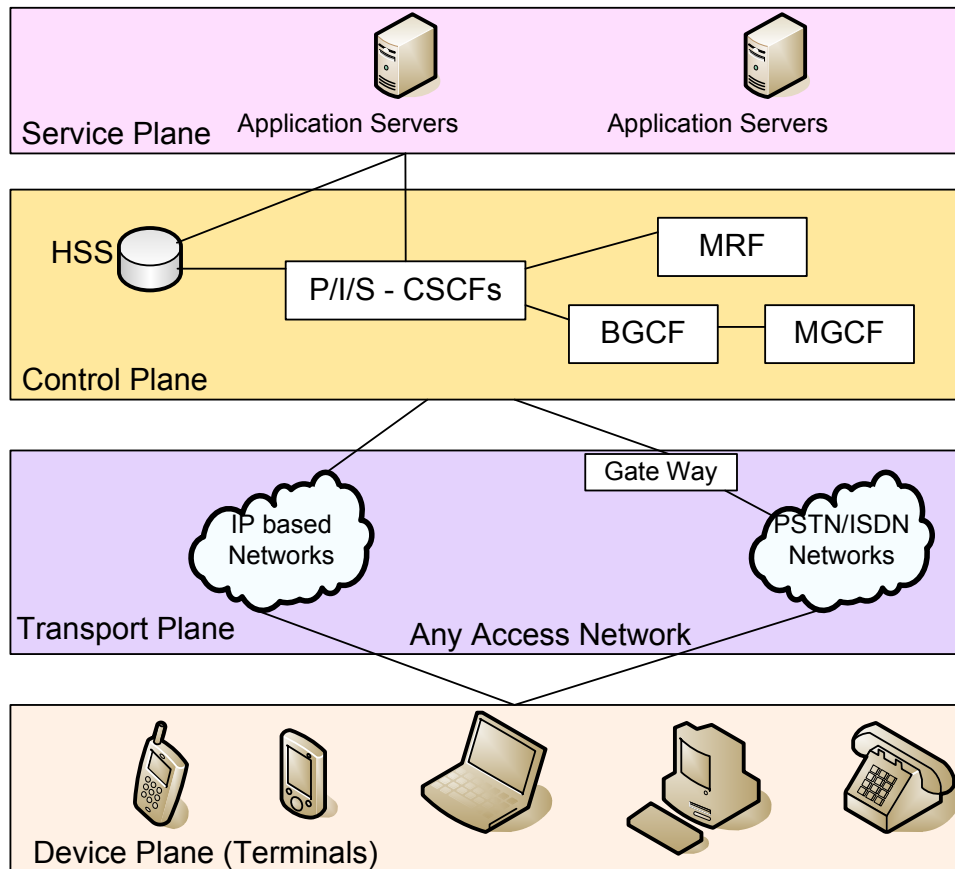


Figure 2.2: The IMS Architecture (Radvision, 2006)

The service plane consists of the Application Servers (AS). All the IMS services run within the ASs and a single AS can handle multiple multimedia services. The AS also provides the interface with the control layers using SIP, such as IMS-Service Control (ISC) interface. Some examples of ASs include the Presence servers, group list management servers, IM servers and conferencing servers.

### 2.3.4 The IMS Network Elements

The IMS overlay architecture is widely abstracted from the air interfaces, hence IMS can be used for any mobile access network technology, as well as for fixed-line access technology, as currently promoted by TISPAN within the NGN reference architecture definition (Magedanz,

Witaszek & Knuettel, 2005). The 3GPP IMS introduces some new elements which require the network operators to upgrade their Core Network (CN) to provide IMS services.

Some of the IMS network elements are shown in Figure 2.3, below.

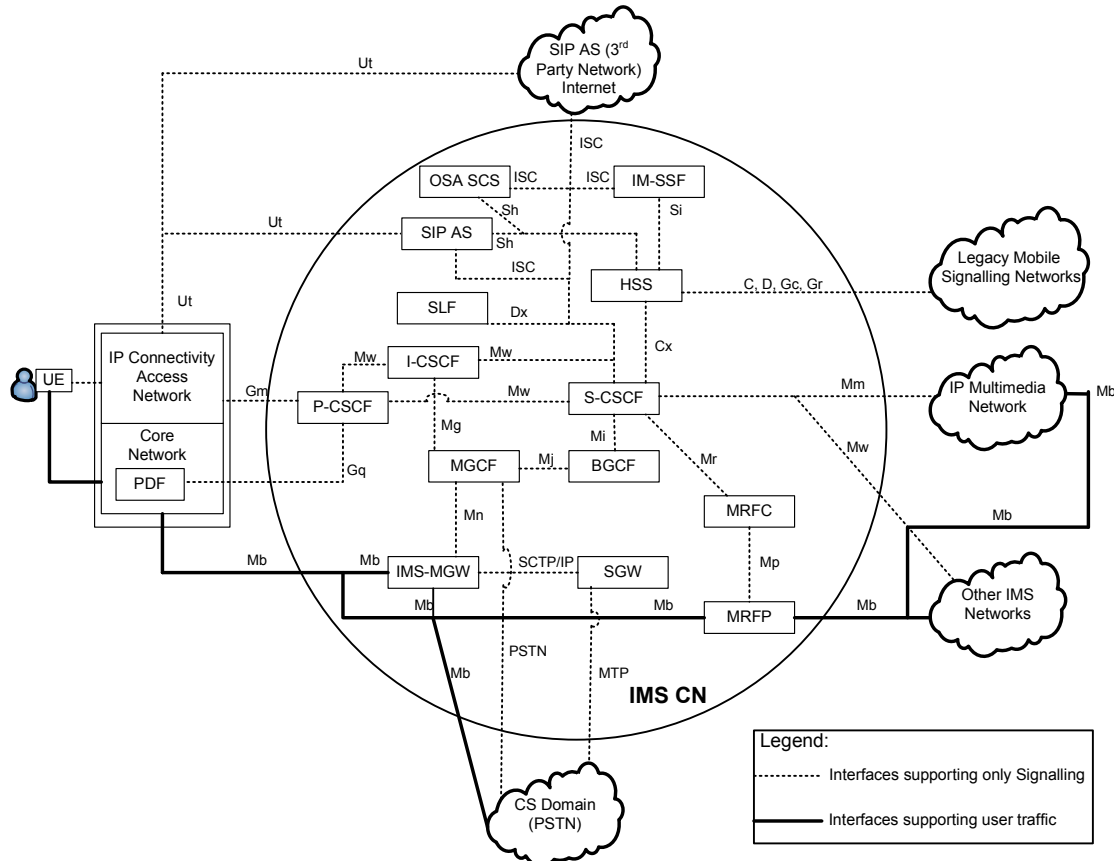


Figure 2.3: The 3GPP IMS Architectural Elements

i. *User Equipment (UE)*

The UE is the IMS-capable terminal used by the subscriber to access IMS services. It contains the SIP UA that generates and terminates the exchange of SIP messages on the user's behalf. Once an IP address has been allocated for registration, the UE cannot change it while engaged in an active dialog.

ii. *Proxy - CSCF (P-CSCF)*

The P-CSCF is the first contact point within the IMS network by the User Equipment (UE). It accepts requests and services them internally or forwards them. The P-CSCF does not modify the request URI in the SIP INVITE message. It can sometimes behave like a UA by terminating and independently generating SIP transactions. The P-CSCF can also generate Charging Data Records (CDR), emergency session detection and also maintain a security association between itself and each UE.

iii. *Interrogating - CSCF (I-CSCF)*

I-CSCF is responsible for querying the HSS to determine the S-CSCF for the user. It is the contact point within an operator's network. It is also responsible for establishing the interface between two different IMS networks such as the home and visitor network. An I-CSCF has the Topology Hiding Inter-network Gateway (THIG) which can be used by the network operator to hide network configuration and topology. This function hides the addresses of operator network entities from being passed outside the operator's network.

iv. *Serving – CSCF (S-CSCF)*

The S-CSCF is the heart of the IMS network. It is responsible for processing registrations, for recording the location of each user and also for performing the user authentication, calls processing and routing of calls to the ASs. The S-CSCF performs session control services for the UE. A network operator may have multiple S-CSCFs with each S-CSCF handling different functions. Other S-CSCF functions includes the interaction with AS for the support of services, forwarding SIP request/response to a BGCF for call routing to the PSTN or CS domain and also generating the CDR.

v. *IMS Media Gateway*

The IMS Media Gateway (IM-MGW) translates between the IP media streams on Time Division Multiplex (TDM) networks. The Media Gateway Control Function (MGCF) controls the media gateway by giving commands that indicate what to convert and when. The MGCF

can take one stream from a Voice over IP (VoIP) network and convert it to one circuit in the TDM network.

*vi. Media Resource Function Processor/ Controller*

The Media Resource Function Controller (MRFC) is responsible for the control of media stream resources in the Media Resource Function Processor (MRFP) and the generation of CDR. It also screens the information coming in from the AS and S-CSCF and then controls the MRFP accordingly. The MRFP is responsible for the control of bearer on the Mb interface and providing the resources to the MRFC.

*vii. Breaking Gateway Control Function*

The Breaking Gateway Control Function (BGCF) is used to dial the CS domain users from the IMS network. It selects the network in which PSTN breakout should occur and receives such requests from the S-CSCF. When a user in the IMS network wishes to communicate with other users in the CS domain, the BGCF forwards the session signalling to the called user. The BGCF also generates the CDRs.

*viii. Home Subscriber Server (HSS)*

The IMS architecture contains two main databases: HSS and the Subscription Locator Function (SLF). While the GSM systems use the Home Locator Register (HLR) to store the user profile, HSS provides the main data storage for all subscriber and service-related data of IMS. The data stored in the HSS includes public and private user identities, registration information, access parameters, service-triggering information, and user-specific requirements for S-CSCF capabilities. The SLF is used in the network operators who have multiple HSS as a resolution mechanism that enables the I-CSCF, S-CSCF and the AS to find the address of the HSS that holds the subscriber data for a given user identity.

*ix. Application Server (AS)*



Service provisioning in IMS is achieved by the AS, which are contacted on the basis of initial filter criteria. Filter criteria are downloaded by the S-CSCF from the HSS during registration and are part of IMS subscribers' service profile (Poikselkä *et al*, 2004: 212). An AS is a special server and a unique service network entity that fulfils a fixed specific role. It is also responsible for detecting the subscriber presence for IM and presence application. The Service Delivery Platform is a server that extends the AS, therefore it can provide more than a specific service. It enables the creation of service on demand.

*x. Policy Decision Function*

Policy Decision Function (PDF) takes a service level policy request from the application layer (for example P-CSCF) and translates it into IP QoS parameters. In General Packet Radio Service (GPRS) networks, the PDF uses the 'Go' interface to set the policy for sessions in the Gateway GPRS Support Node (GGSN). In TISpan-based networks, the PDF contacts the Border Gateway Function (BGF) to enforce the policy. PDF and P-CSCF were combined in Release 5 (3GPP TS 24.228, 2006) specification of the 3GPP IMS, but the PDF was separated, as a stand alone element, from P-CSCF starting from Release 6 specification of the IMS.

### **2.3.5 The IMS Concept and Procedure**

From a technical point of view, the 3GPP IMS is an IP-based overlay network that spans access and core network and provides for IP-based signalling reusing IETF protocols such as SIP and Diameter, both of which are enhanced by specific IMS extensions (Fabini *et al*, 2006). The IMS utilizes the PS domain to transport multimedia signalling and bearer traffic.

To present the concept and procedures used in IMS, the 3GPP Universal Mobile Telecommunication System (UMTS) mobile network will be used. In the GPRS/UMTS network, there are three steps that the UE must perform before the IMS services can be accessed, namely primary Packet Data Protocol (PDP) context activation, P-CSCF discovery and application level IMS registration.

i. *GPRS Packet Data Protocol (PDP) Context Activation*

PDP context is established to enable the UE to transfer data between the PS domains of the GPRS/UMTS network. The PDP is a network protocol used by external packet data network to interface to GPRS (Wong & Varma, 2003), as specified in section 6.1.3.1 of the 3GPP Release 99 (3GPP TS 24.008, 2004). Since the IMS information is delivered through the GPRS transport network, a UMTS mobile station must activate GPRS PDP context before it can register to the IMS network (Lin *et al*, 2005). Figure 2.4 shows the UE initiated PDP context activation.

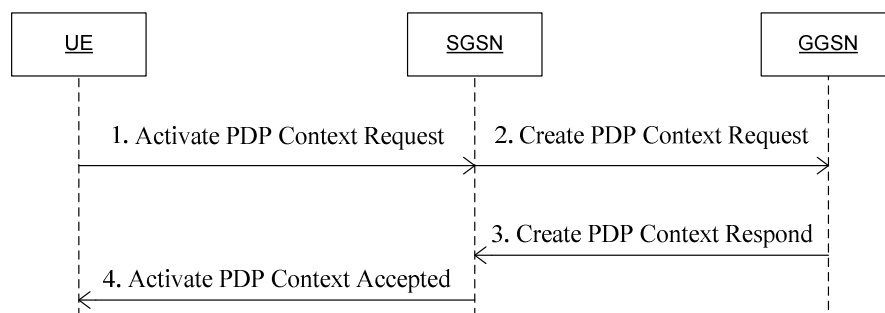


Figure 2.4: UE Initiated PDP Context Activation (Pouse *et al*, 2003)

ii. *Local P-CSCF Discovery*

After having activated the primary PDP context, the second step is to discover the local P-CSCF before the user can register on the IMS network. The discovered P-CSCF will be the first contact point for the UE in the IMS network. The 3GPP suggested two methods that can be used by the UE to discover the P-CSCF (3GPP TS 23.228 and 3GPP TS 24.228); otherwise the P-CSCF can be pre-configured within the UE. Dynamic Host Configuration Protocol (DHCP) and the Domain Name System (DNS) can be used to resolve the P-CSCF domain name and/or IP address as specified in RFC 3319 (Schulzrinne & Volz, 2003). This is shown in Figure 2.5 below.

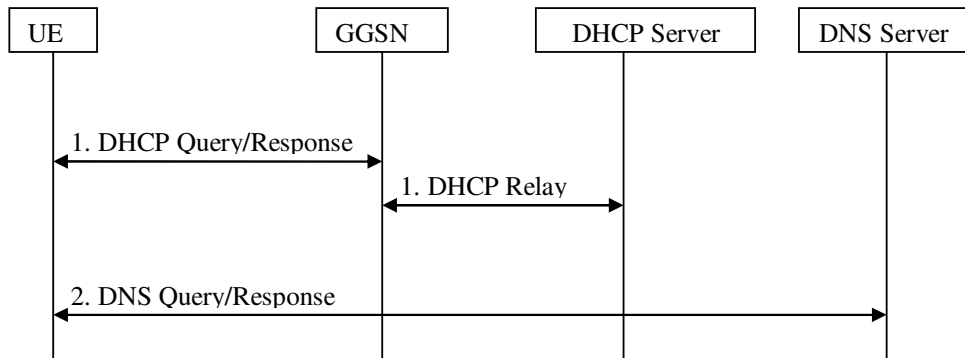


Figure 2.5: P-CSCF Discovery using DHCP Procedure (3GPP TS 23.228, 2006)

iii. *Application Level SIP Registration*

The third and final step in accessing the IMS services is the UE application level SIP registration. Registration creates bindings in a location service for a particular domain that associates an address-of-record (AOR) Uniform Resource Identifier (URI) with one or more contact addresses (Rosenberg *et al*, 2002). The UE uses the same registration procedure for registering on the home or visited network. And it can also register multiple public identities through a single IMS registration procedure. Appendix A.1 depicts the message flow for the application level registration.

Once the above three steps have been performed successfully, the UE will be ready to establish a SIP session to access IMS services. Figure 2.6 shows the routing of an INVITE request from UE A (originator) to the UE B (destination).

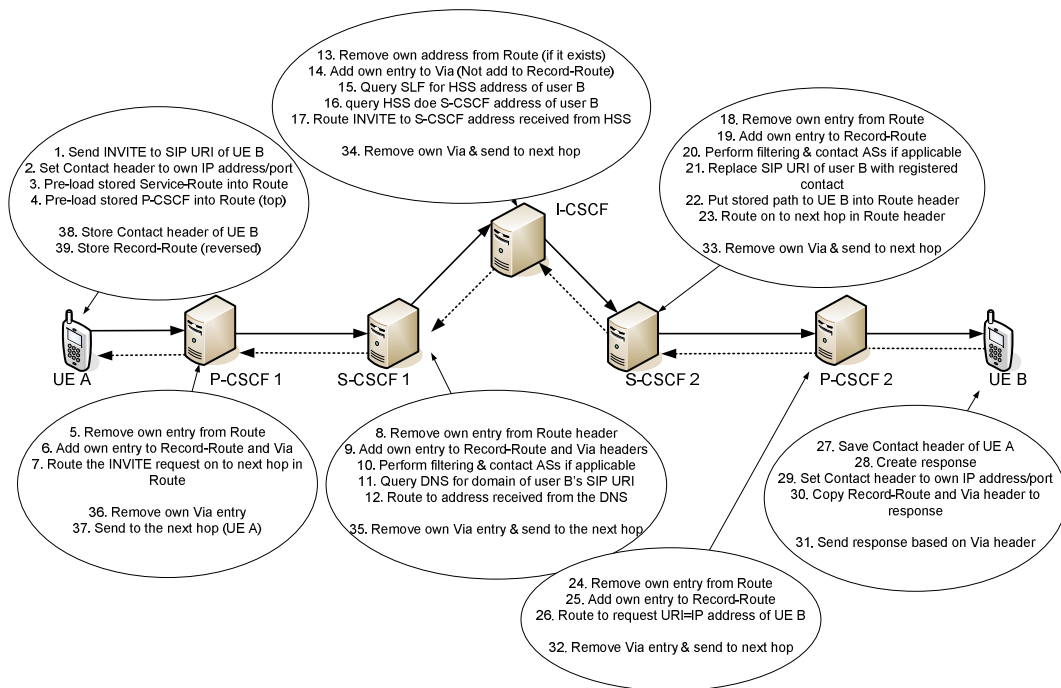


Figure 2.6: INVITE routing (adapted from Poikselkä *et al*, 2004)

The originating user can establish a session by sending an SIP INVITE request to the intended destination. Appendix A.2 depicts the message flow for the session establishment, while Appendix A.3 depicts the session termination message flow.

### 2.3.6 User Identification in IMS

The home network operator assigns public and private user identities to all subscribers belonging to their IMS network. The private user identity is used for registration, authorisation, administration and accounting purposes. The private user identity is in the form of a Network Access Identity (NAI) as defined in RFC 2486: [username@operator.com](http://www.ietf.org/rfc/rfc2486.txt). The private user identity information is securely stored in an International Mobile Subscriber Identifier (ISIM) application and is allocated to the user permanently. A private user identity performs a similar function in IMS as an International Mobile Subscriber Identifier (IMSI) does in GSM (Camarillo & García-Martín, 2006:41). The format for the private user identity is derived from the IMSI and is specified in the 3GPP TS 23.003.

The IMS public user identity can be an SIP URI, as defined in RFC 3261 or a Tel URL as defined in RFC 2806, and is allocated to the users by their network operator. The IMS subscribers use this public user identity to be identified by other users. This identity is stored in the ISIM and it's authenticated by the home network during registration. The public user identity is equivalent to the Mobile Subscriber ISDN Number (MSISDN) in the GSM system.

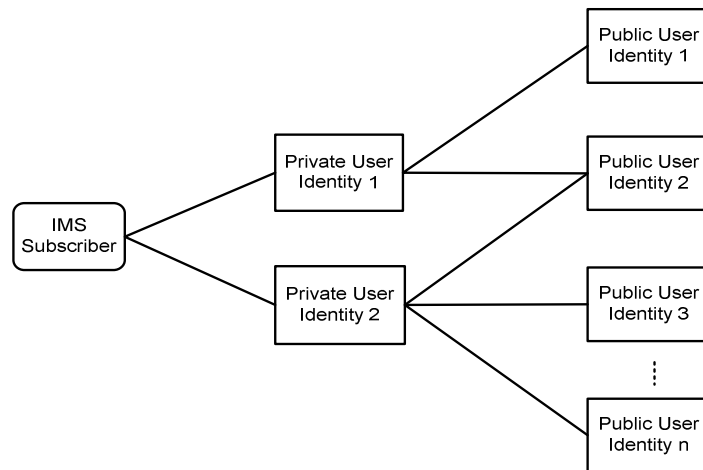


Figure 2.7: Release 6 user identities (Camarillo & García-Martín, 2006:43)

Release 6 of 3GPP has extended the relationship between private and public user identities by allowing a single subscriber to have more than one private user identity. This relationship is shown in Figure 2.7. In the case of UMTS, only one private user identity can be stored in the Universal Integrated Circuit Card (UICC), although users may have different UICC inserted in different IMS terminals. The UICC is a removable smart card that contains a limited storage of data.

The IP multimedia Services Identity Module (ISIM) which is contained within the UICC is standardized by 3GPP (3GPP TS 31.103, 2005). ISIM is very important for IMS because it contains the collection of parameters that are used for user identification, user authentication and terminal configuration when the terminal operates in IMS (Camarillo & García-Martín, 2006:45). The following parameters are stored within the ISIM:

- *Private User Identity*: one Private User Identity can be stored in ISIM
- *Public User Identity*: one or more Public User Identities are allocated to users

- *Home Network Domain URI*: is SIP URI containing the home network domain name.
- *Long-term Secret*: this is used for authentication purposes and for calculating the integrity and cipher keys.

### 2.3.7 IMS in the Next Generation Networks

IMS uses Internet technologies and protocols in order to allow users to move across countries while still being able to execute all the services as if they are operating from their home networks (Chen *et al*, 2006). Though IMS was originated by the 3GPP for mobile networks, the ETSI showed interest in IMS by incorporating it in their TISPAN NGN standard (ETSI TR 180.001, 2006). Besides ETSI-TISPAN, other wireline standardization bodies have also adopted the IMS as their NGN core.

While TISPAN tackles non-SIP applications the 3GPP IMS does not, because non-SIP applications include not only legacy services, but also Hypertext Transport Protocol (HTTP) applications and peer-to-peer file sharing (Hills, 2006). The process of fitting IMS into ITU's NGN is performed by the ETSI TISPAN group which is responsible for all aspects of standardization for present and future converged networks (Fabini *et al*, 2006). Figure 2.8 below shows the TISPAN IMS elements. Items shown in green are associated with PSTN egress and ingress, while items in burgundy are associated with IP to IP SIP calls.

The Interconnect Border Control Function (IBCF) was introduced by TISPAN to formalize interconnectivity between networks. Its roles include the provision of Network Address and Port Translator (NAPT) and firewall functions for signalling, policing of signalling, topology hiding and conversion between IPv4 and IPv6. Through the control of a BGF, the IBCF also controls the media exchanged across the operator boundary, controlling media pinhole establishment, policing, and dynamic NAPT and firewall functions (Newport-Networks, 2005).

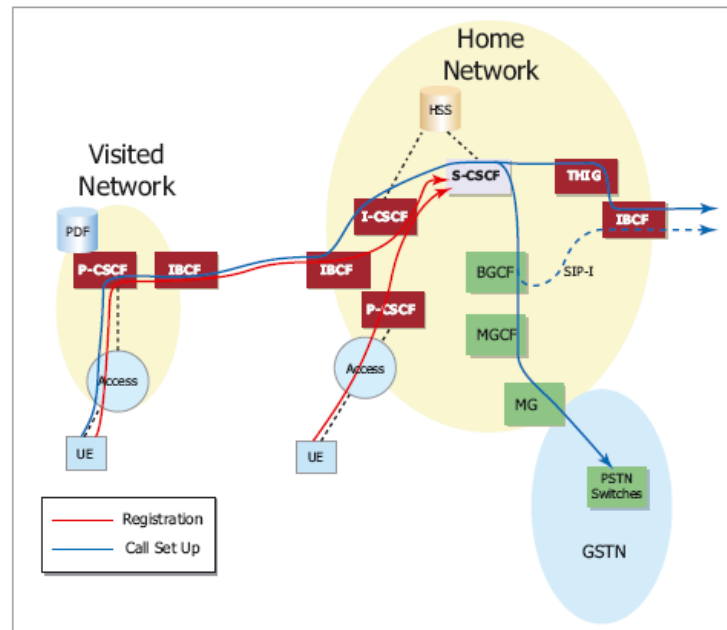


Figure 2.8: TISpan IMS Functional Elements (Newport-Networks, 2005)

## 2.4 SESSION INITIATION PROTOCOL – SIP

SIP is an application layer signalling protocol developed by the IETF to set up, modify and tear down multimedia sessions over the IP network (Rosenberg *et al*, 2002). SIP is a close relative of HTTP 1.1 and Simple Mail Transport Protocol (SMTP) protocols. From HTTP, SIP borrowed a client-server design and the use of Uniform Resource Locators (URL) while from the SMTP it borrowed a text-encoding scheme and header style (Johnston, 2001:1). This makes SIP a textual request-response protocol. SIP presents a revolution in communications by abandoning the telecom signalling and control models developed for telephony over many years in favour of Internet and web-based protocols (Camarillo, 2002).

In this section the SIP background and history, the SIP architecture, network and the SIP message structures are discussed.

### 2.4.1 SIP Background and History

SIP originated in late 1996 as a component of the IETF multicast backbone (Mbone). Mbone is an experimental multicast network on top of the public Internet. Due to its simplicity and

extensibility, SIP was adopted as a VoIP signalling protocol as an IETF-proposed standard in 1999 and published under the RFC 2543. SIP was further enhanced to take into account interoperability functions, better design and new features. In 2002 RFC 3261 was released as the core proposed standard, this is backward compatible with RFC 2543. RFC 3261 is currently the core SIP specification as defined by IETF (Rosenberg *et al*, 2002). SIP became more widely used for VoIP services than H.323 multimedia protocol. Today, SIP is at the heart of the IMS network architecture, providing the real-time, peer-to-peer, multiparty and multimedia capabilities of IMS (BEA, 2006).

## 2.4.2 SIP Architecture

SIP fits within the existing Internet multimedia protocol stack as shown in Figure 2.9 below. SIP elements can be classified as User Agents (UA), proxies and servers or intermediaries. Though it is possible for UAs to communicate directly without the server, servers are generally used by network administrators and service providers to keep track of traffic in their network, and also to protect their business.

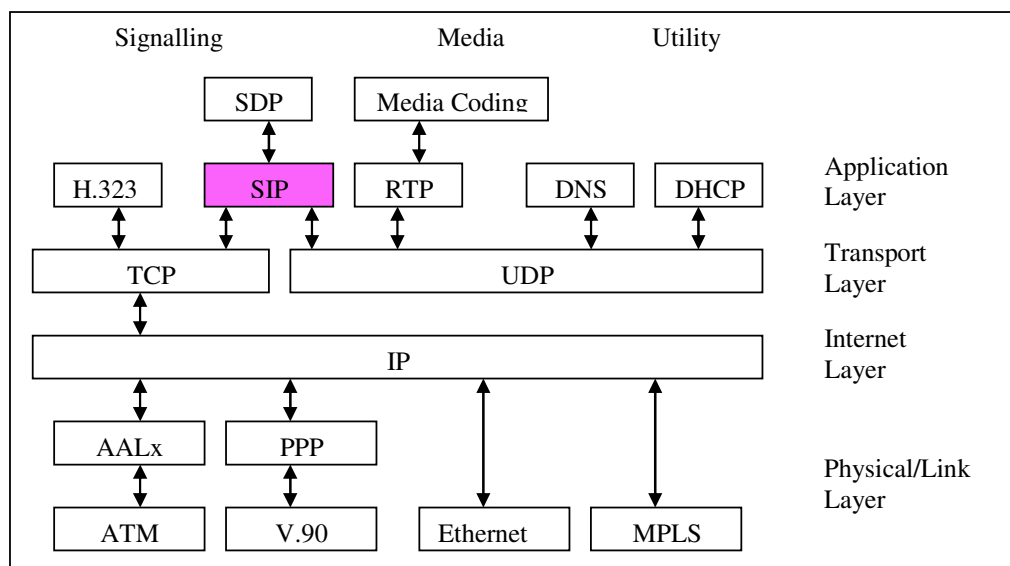


Figure 2.9: The Internet multimedia protocol stack (Johnston, 2001:4)

Shown in Figure 2.10 are the SIP network elements. An SIP UA is the endpoint for dialogs.



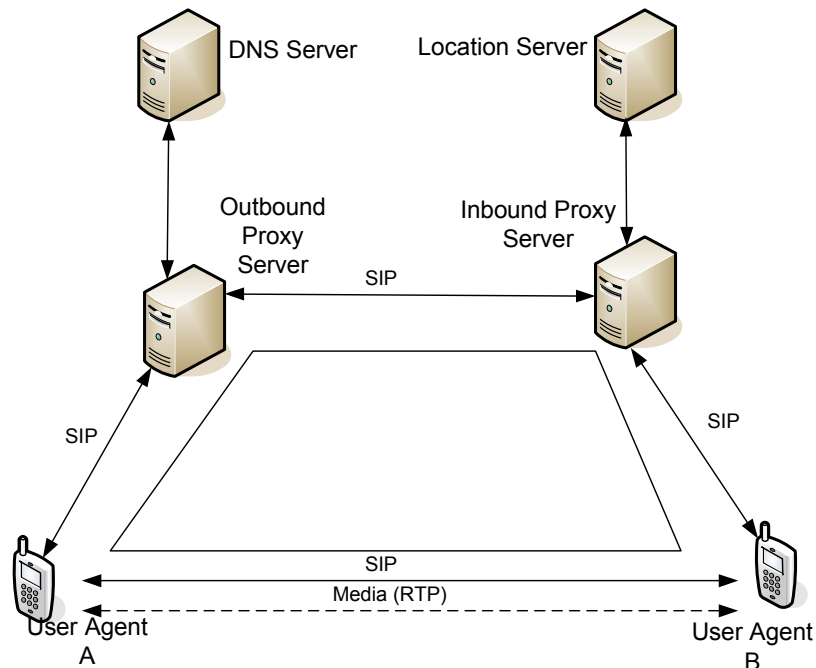


Figure 2.10: SIP Trapezoid (Poikselkä *et al*, 2006)

UA runs within a UE and is divided into two parts: UA Client (UAC) and UA Serve (UAS) (Rosenberg *et al*, 2002). SIP intermediaries are logic entities through which SIP messages pass on their way to their final destination. They are used to route and redirect the requests. In practice, these entities are owned or housed within the network operator’s premises.

### 2.4.3 SIP Messages Format

The SIP message is made up of three parts as shown in Figure 2.11. The start line contents vary depending on whether the SIP message is a request or a response. If the message is a request, it is called a request line. If the message is a response, the start line is called a status line.

The request line has three components:

- Method – there are six base SIP request methods (see Table 2.1). Other methods were created as extensions to RFC 3261 (see Table 2.2).
- Request-URI – is a SIP or Secure SIP (SIPS) URI that identifies a resource that the request is addressed to.

- Protocol version - all requests compliant with RFC 3261 must include version 2.0 in the request, in the following form: “SIP/2.0”.

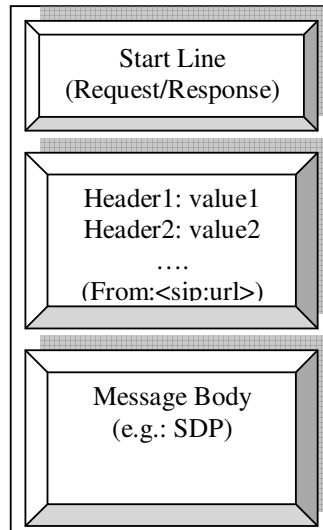


Figure 2.11: SIP message format (Adapted from: Rosenberg *et al*, 2002)

The status or response line consists of three components:

- Protocol version – identical to the protocol version in the request line.
- Status Code – a three-digit code that identifies the nature of the response.
- Reason Phrase – a free text field providing a short description of the status code, that is mainly aimed at human users.

There are two types of SIP responses: provisional/informational (1xx) and final response (2xx to 6xx). These responses are grouped into status codes, of which there are six classes:

- 1xx – provisional or information responses. They indicate that the request has been received and the recipient is processing the request
- 2xx – success response.
- 3xx – re-direction responses. The requester needs to take further action to complete the request.
- 4xx – client-error responses
- 5xx – server-error responses

- 6xx – global-failure responses. The request cannot be fulfilled at any server.

#### 2.4.4 SIP Request Methods

The IETF defined about six methods used in RFC 3261 for different types of SIP request, as shown in Table 2.1.

Table 2. 1: Six RFC 3261 SIP Request Methods

Method	Description
<b>ACK</b>	Confirms that the client has received a final response to a request.
<b>BYE</b>	Used by a client to tell the server that it wishes to release the call.
<b>CANCEL</b>	Cancels a pending request with the same Call-ID, To, From and Call sequence number (Cseq) header field values.
<b>INVITE</b>	Indicates that the user or service is being invited to participate in a session. The message body may contain a description of the session.
<b>OPTIONS</b>	Queries the capabilities of the other side
<b>REGISTER</b>	Used by a client to register the address listed in the “To” header field with a SIP server.

In addition to the original six SIP methods, other methods are defined by the IETF as the extensions to the RFC 3261 SIP. These methods are shown in Table 2.2.

Table 2. 2: Other SIP Request Method Extensions

Method	RFC #	Description
<b>INFO</b>	RFC 2976	Transfers information during a session
<b>MESSAGE</b>	RFC 3428	Allows the transfer of IM.
<b>NOTIFY</b>	RFC 3265	Informs the user about the subscribed event.
<b>SUBSCRIBE</b>	RFC 3265	Enables the user to subscribe to certain events
<b>REFER</b>	RFC 3515 RFC 4488	Enables the sender of the request to instruct the receiver to contact a third party.
<b>PRACK</b>	RFC 3262	Provisional Reliable ACK (PRACK) plays the same role as ACK request, but for provisional responses.

		(Rosenberg & Schulzrinne, 2002)
<b>UPDATE</b>	RFC 311	Allows a client to update parameters of a session, but has no impact on the state of a dialog. (Rosenberg, 2002)
<b>PUBLISH</b>	RFC 3903	PUBLISH is similar to REGISTER in that it allows a user to create, modify and remove state to another entity which manages this state on behalf of the user.

#### 2.4.5 SIP P-Headers for IMS

Beside the use of SIP extension methods, there are several SIP Private-headers (P-headers) used in 3GPP IMS. Marshall (2003) defined RFC 3313 which introduces the P-Media-Authentication header field that contains one or more media authorization tokens which are to be included in subsequent resource reservations for the media flows associated with the session. RFC 3325 problem introduces private extensions to the SIP, that enable a network of trusted SIP servers to assert the identity of authenticated users, and to apply existing privacy mechanisms to the identity problem (Jennings *et al*, 2002). RFC 3455 introduced a set of private SIP headers used by the 3GPP along with their applicability, which is limited to particular environments (Garcia-Martin, Henrikson & Mills, 2003). In the 3GPP-IETF collaboration, work is still going on to extend the core RFC 3261 SIP specification to make the IMS a fully IP based system.

#### 2.5 PERSONAL NETWORKS – PN

A PN concept extends the local scope of Personal Area Network (PAN) to a global network that creates a virtual personal environment which is capable of spanning a variety of infrastructure-based and ad-hoc networks (Calin *et al*, 2006). As more customers own more than one networked terminal (such as mobile phone, Personal Digital Assistance, or a laptop) which may be either in close vicinity or anywhere in the world, they would like to interact with them at any time. It is impractical for customers to carry all these gadgets with them at all times, but with the PN, customers are able to manage or interact with all their networked devices wherever they are. The federation of PNs on the other side allows the group of people

with their own PNs to interact and share information. Service discovery plays an important role in the successful utilization of PNs.

One can refer to a PN as a dynamic collection of interconnected heterogeneous personal devices, as depicted in Figure 2.12. The networking capability of most heterogeneous devices owned by a person, enables him/her to have flexible network solutions to access their personal devices anywhere and at any time. PN requires more services than PAN because its devices are separated by hundreds of kilometres, and its connection technologies include medium to national and continental range such GPRS or UMTS (Prasad & Skouby, 2005). The PN is the interconnection of PANs with Wireless-Wide Area Network (W-WAN) to facilitate access to many services available through these networks, plus the interconnection to the federation of PNs. The PN ranges across spheres defined around a person, because it starts at the PAN sphere and extends to the immediate environment, the instant partners, radio access and the interconnectivity sphere to the cyber world sphere (Charas *et al*, 2000).

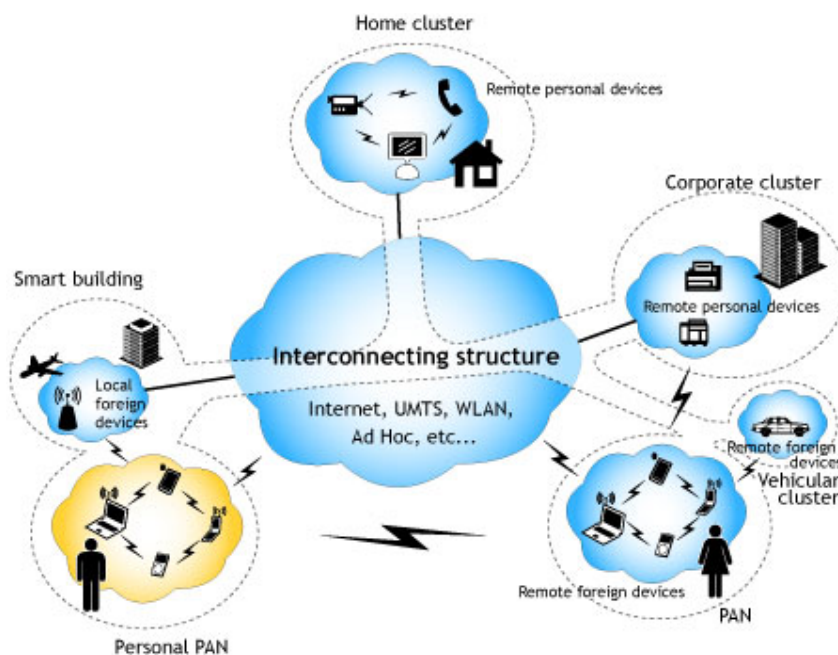


Figure 2.12: Personal Network (MAGNET, 2006)

## 2.6 INTRODUCING THE MAGNET BEYOND PROJECT

MAGNET, which stands for My Adaptive Global Network is an integrated project supported within the Sixth Framework Programme (FP6) of the EU Commission. MAGNET's long-term vision is to ensure that future users will be supported in their private and professional activities by their own PN. This consists of a core PAN extended with clusters of remote devices, which can be shared, public or private, and adaptable to the quality of the accessed network (Calin *et al*, 2006). MAGNET Beyond is divided into seven work packages (WP), which together will meet its research objectives.

This study was carried out by the department of wireless networks and multimedia services at INT-Evry in collaboration with TUT-FSATIE. INT was involved in the MAGNET Beyond project WP 2. WP 2 focuses on the PN networking. The objectives of the department in WP 2 is provision for interoperability of Universal Plug and Play (UPnP) and web services as a non-negligible overlay service discovery and provisioning framework, to be expected in current and future networks. This will cover interaction of PN service framework with external service architectures such as IMS, SIP or other IP-based services.

## 2.7 RELATED WORK

Since the introduction of IMS in 2002, continuous research work has been conducted in the development of IMS-capable terminals and network elements. Bublely (2006) conducted a research into the evolution of SIP and IMS handset. “*While much attention has focused on deployments of IMS network infrastructure and applications, the need for a new class of phones has been largely forgotten*” (Bublely, 2006). As shown in Figure 2.13, mass market penetration of IMS functionality onto handsets will only be attained in 2009, and in most cases this will be through *partial IMS* implementations.

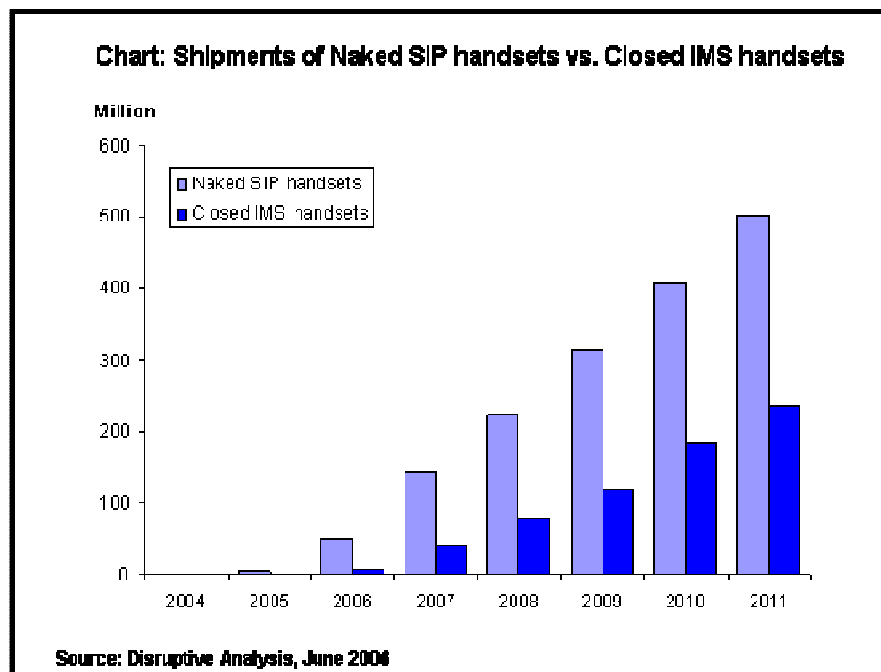


Figure 2.13: Shipments of Naked SIP versus Closed IMS handsets (Bubley, 2006)

Currently there are three open IMS Clients available to the research community. Of these three IMS Clients, only one does not have free source code for download as it is just open for use. The other two are open in the sense that developers can download their source codes and extend them. Although these IMS Clients have not targeted the handheld mobile devices, one of them (FOKUS OpenIC Lite) can be deployed in some PDAs.

### 2.7.1 FOKUS Open IMS Client Lite

The FOKUS group has developed an IMS client which they call Open IMS Client (OpenIC) Lite, as shown in Figure 2.14. Work on FOKUS OpenIC Lite is progressing towards a robust IMS client with diverse services for Fixed Mobile convergence (FOKUS, 2007). The solutions delivered by the framework align with IETF, 3GPP and TISPAN client specifications. The client is available in Java and in .NET and runs on multiple platforms (Windows Mobile, Linux and Windows XP).

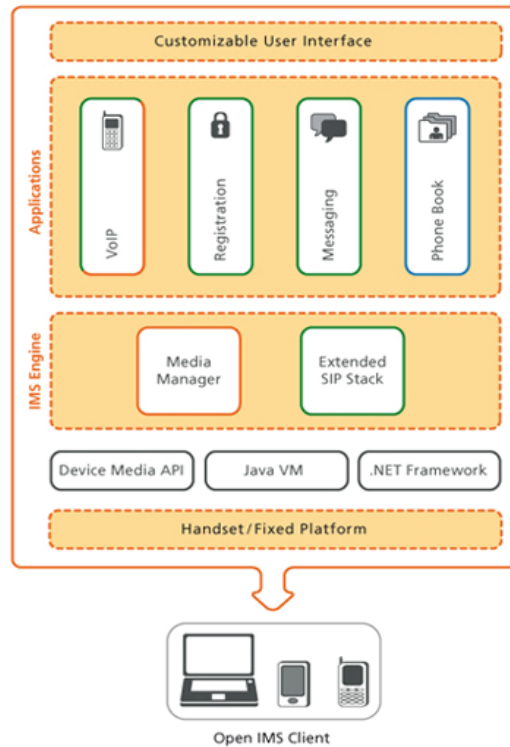


Figure 2.14: FOKUS OpenIC Lite (FOKUS, 2007)

While OpenIC Lite is free and available for download, the FOKUS group has a commercial Open IMS Client version called OpenIC. "Open" in this sense means extendable therefore it should not be confused with "Open Source" (FOKUS, 2007).

### 2.7.2 IMS-Communicator

IMS-Communicator is an open source project that was originally developed by PT Inovação to support the development and testing of IMS and NGN components for its Service Handling on IP Networks (shipnet®) IMS/NGN architecture (PT Inovação, 2007). The IMS-Communicator, shown in Figure 2.15, is based on the old version of the SIP-Communicator soft-phone project. It is built on top of the JAIN-SIP reference implementation, and makes use of the Java Media Framework (JMF) API as its media stack.





Figure 2.15: IMS-Communicator Screenshot (PT Inovação, 2007)

According to the IMS-Communicator website, the developers of IMS-Communicator have no plans for porting it to the J2ME Connected Device Configuration (CDC) environment. As a result, this IMS Client cannot be deployed to mobile devices. The IMS-Communicator project is licensed by the Apache Software License and the GNU Lesser General Public License (LGPL). Its hosting is provided by the BerliOS developers.

### 2.7.3 UCT IMS Client

University of Cape Town (UCT) IMS Client is another open source IMS Client hosted by the BerliOS developers. The UCT IMS Client has been developed by the communications research group at UCT in South Africa, and it is designed to be used in conjunction with the Fraunhofer FOKUS Open IMS Core network testbed. On the UCT website the following is stated about the UCT IMS Client “This project is still in active development and there are many known bugs”. The UCT IMS Client screenshot is shown in Figure 2.16. The UCT IMS Client has been developed in C++ language and is only available for Linux operating systems. It is not available for mobile devices.

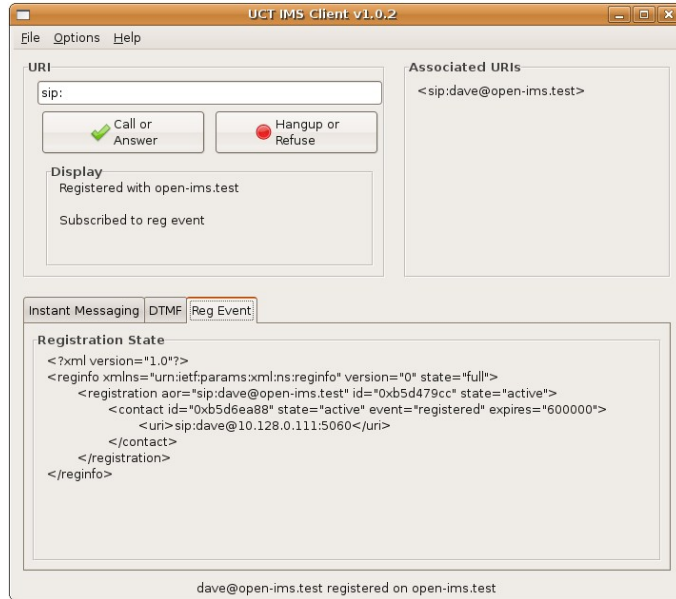


Figure 2.16: UCT IMS Client Screenshot (Waiting & Good, 2007)

## 2.8 LIGHT-WEIGHT IMS CLIENT CONCEPT

The three open IMS Clients discussed in Section 2.7, above, were mainly targeted to be implemented within the end user devices that have high processing power and memory. These end user devices may include laptops, desktop personal computers or Personal Digital Assistance (PDA). Hence the three open IMS Clients may not be deployed within the current CLCD based handheld mobile devices.

Based on the 3GPP IMS, the IMS compliant end user has to provide the necessary SIP signalling support, the service related media codecs for multimedia applications in addition to the basic connectivity support, such as IP (Magedanz, Witaszek & Knuettel, 2005). Current open IMS Clients do provide these supports to the end users with higher processing power and memory devices.

In this study, the term *Light-Weight* is used to define the IMS Client that is developed under the Java Micro-Edition (J2ME) platform. Though developed within the J2ME platform, this client should be capable of providing all the IMS supports as specified by the 3GPP and IETF

standards and recommendations. As a result, this study aims to developing an IMS Client that can be deployed within the J2ME-compliant handheld mobile devices.

## **2.9 CONCLUSION**

In this chapter, a literature survey of the IMS, SIP and PN was presented. IMS defines a complete architecture and framework that enables the convergence of wireline and mobile network technologies over an IP-based infrastructure. SIP is an IETF signalling protocol for initiating, managing and terminating multimedia sessions across packet networks. The PN is an extension to a local scope of the PAN. This chapter also introduced the MAGENT Beyond project and the involvement of this study with the MAGNET Beyond project. Chapter 3 reviews Java 2 Micro-Edition (J2ME) as the tool for the development of portable devices.

## **CHAPTER 3**

### **TOOLS FOR DEVELOPING JAVA BASED PORTABLE APPLICATIONS**

#### **3.1 INTRODUCTION**

The development of the IMS Client in this project was based on Java 2 Micro-Edition (J2ME) platform. The development software and hardware tools used for rich application portable devices are presented in Section 3.2. Section 3.3 introduces and discusses the J2ME as the chosen development platform for the IMS Client, while Section 3.4 presents the Java libraries or Application Programming Interfaces (APIs) used to support the SIP signalling.

#### **3.2 DEVELOPMENT TOOLS**

Different software and hardware tools and libraries are available for use by developers to develop rich applications for portable devices such as mobile phones, set-top boxes, PDAs and navigation systems. Mao, Talevski & Chang (2007) have compared the most popular three mobile development platforms, and presented their results as shown in Table 3.1. Due to its extensive market penetration and developer community support, J2ME development platform was chosen for the development of the IMS Client.

##### **3.2.1 Java Development Kit (JDK)**

The JDK consists of a Java compiler, written in Java, and a run-time interpreter for a particular platform. It can be downloaded and installed free from Sun website. For the development of the IMS Client, in this study, jdk 1.5 was used.

Table 3.1: Mobile Development Platforms (Mao, Talevski & Chang, 2007)

<b>Functionality</b>	<b>Symbian</b>	<b>J2ME</b>	<b>BREW</b>
Foundation	C++	Java	C++
Learning Curve	Difficult	Average	Difficult
Emulator	Free	Free	Limited
Debuggers Available	Latest version– good	Excellent	Needs payment
Development Tool Cost	Varies	Free	Expensive
Cross-Platform Deployment	Compile per target	Average	CDMA handsets only
Developer Community and Support	Extensive	Extensive	Limited
Market Penetration	Extensive	Extensive	In few countries
SIP Support	Yes	Yes	Unknown
RTP Support	Yes (complicated)	No	Unknown

### 3.2.2 J2ME Sun Wireless Toolkit

The Sun Wireless Toolkit (WTK) is the reference implementation of J2ME. It has building tools, utilities and a device emulator for creating J2ME Java applications which are compliant to Mobile Information Device Profile (MIDP). The Sun WTK provides the byte-code pre-verification tool, implementation of API class libraries and a smart-phone device emulator (Goyal, 2005). KToolbar is the main Sun WTK component which is used to build MIDlets, to launch the emulator, and to start the utilities provided by the Sun WTK. Java applications that run in the MIDP environment are called MIDlets. The IMS Client is developed as a MIDlet and emulated using the Nokia smart-phone emulator provided with the Sun WTK version 2.5. Version 2.5 of the Sun WTK is compliant with Java Technology for the Wireless Industry (JTWI) JSR 185 and Mobile Service Architecture (MSA) JSR 248 specifications for mobile devices, as shown in Figure 3.1.

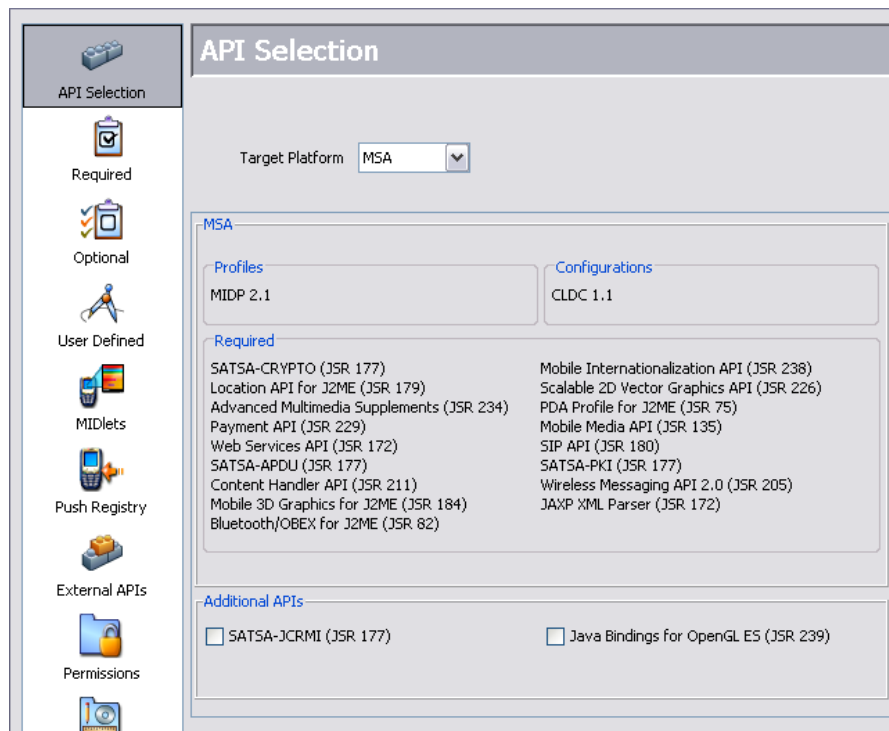


Figure 3.1: Sun Java WTK 2.5 API Selection Window

From the API Selection window, one can decide whether to use the JTWI or MSA target development platform. Many of today's Java enabled wireless mobile devices are mainly based on Connected Limited Device Configuration (CLDC) and MIDP; with the latest mobile devices being compliant with Java MSA API (JCP JSR 248, 2006). Such devices include Nokia N81, N91 and Sony-Ericsson Z750.

### 3.2.3 EclipseME

Although the Sun WTK provides a powerful set of J2ME development tools, it does not offer the Integrated Development Environment (IDE) experience. For developers who are used to the convenience of IDE, the Sun WTK lacks key productivity features, such as team collaboration, context-sensitive code editors, debugging and project management. In order to provide the IDE experience to developers, the Eclipse IDE can be extended with the EclipseME plug-in to enable the integration between the Eclipse and the Sun WTK.

In order to integrate both the Eclipse IDE and Sun WTK, both tools must be downloaded and installed in the same machine. The EclipseME plug-in makes use of the Sun WTK emulator during runtime, while Eclipse IDE is used as a text editor.

### **3.3 JAVA 2 MICRO-EDITION PLATFORM**

Java was introduced in 1995, but its platform capabilities took off with the advent of servlets. Servlets are Java programs that run on a server, offering a modular and efficient replacement for Common Gateway Interface (CGI). The first revolution of Java occurred when it expanded into the server side with Java 2 Platform, Enterprise Edition (J2EE). J2ME is the second revolution in Java that targets small, mobile devices.

Today, the Java language defines three platforms: Java 2 Platform, Standard Edition (J2SE), J2EE and J2ME. J2SE is designed for desktop computers. Most often it runs on Linux, Solaris or Microsoft Windows operating systems. J2EE is a comprehensive platform for multi-user, enterprise-wide applications. It is based on J2SE and incorporates APIs for server-side computing. J2ME is a set of technologies and specifications developed for small devices like pagers, mobile phones and set-top boxes. J2ME uses subsets of J2SE components, such as smaller virtual machines and leaner APIs.

J2ME is divided into configurations and profiles, as well as optional APIs, which provide specific information about APIs and different families of devices. In the following subsections the structure of J2ME is discussed.

#### **3.3.1 Configurations**

A configuration is designed for a specific kind of device based on memory constraints and processor power. It specifies a Java Virtual Machine (JVM) that can be easily ported to devices supporting the configuration. Device manufacturers are responsible for porting a specific configuration to their devices. Currently there are two configurations, Connected Device Configuration (CDC) and the Connected, Limited Device Configuration (CLDC). The configurations and profiles of J2ME are generally described in terms of their memory capacity

(Li & Knudsen, 2005). The minimum amounts of volatile and non-volatile memory are specified.

*i) Connected Device Configuration (CDC)*

A CDC device has a minimum Read-Only Memory (ROM) of 512 KB, Random Access Memory (RAM) of 256 KB and some kind of network connection (Li & Knudsen, 2005). It is designed for devices like TV set-top boxes, car navigation systems and high-end PDAs. The CDC specifies that a full JVM must be supported.

*ii) Connected, Limited Device Configuration (CLDC)*

CLDC is designed for smaller devices than those targeted by the CDC. Such devices include mobile phones, pagers and PDAs. The CLDC devices have limited display size, limited memory, limited CPU power, limited input, limited battery life and limited network connection. CLDC devices have 160 KB to 512 KB of total memory, including a minimum of 160 KB of ROM and 32 KB of RAM available for the Java platform (Li & Knudsen, 2005).

The reference implementation of the CLDC is based on a small JVM called the KVM. The name derives from the fact that it is a JVM of which the size is measured in kilobytes rather than megabytes (Li & Knudsen, 2005). The KVM cannot do everything a JVM does in the J2SE world because of it is too small.

### **3.3.2 Profiles**

A Profile is based on a configuration and provides additional APIs, such as user interface, persistent storage and whatever is necessary to develop running applications for the device. It is layered on top of a configuration, adding the APIs and specifications needed to develop applications for a specific family of devices (Li & Knudsen, 2005:2).



The Foundation Profile is a specification for devices that can support a rich networked J2ME environment, as depicted in Figure 3.2. It does not support a user interface. Other profiles can be layered on top of it to add user interface support and other functionality.

The PDA Profile (PDAP) is designed for palmtop devices with a minimum of 512 KB combined ROM and RAM, and a maximum of 16 MB. It includes an application model based on MIDlets, but uses a subset of the J2SE Abstract Windowing Toolkit (AWT) for graphic user interface (GUI).

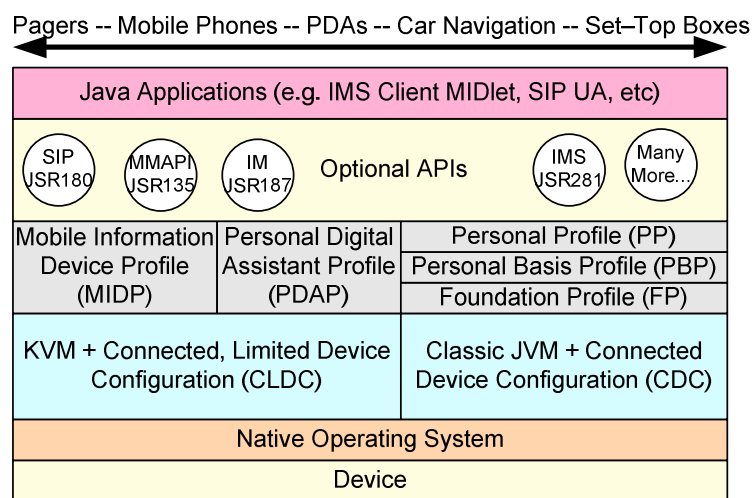


Figure 3.2: J2ME Stack (Li & Knudsen, 2005: 2)

MIDP complements the CLDC by minimizing both the memory and power required for limited devices. It also provides the basic API that is used for creating applications for these devices. The Java Community Project (JCP) has released two versions of MIDP JSR so far (MIDP 1.0 as JSR 37 and MIDP 2.0 as JSR 118). The JSR-248 MSA is a J2ME platform umbrella specification that defines the next step in the Java platform evolution for mobile handsets, based on the CLDC (JCP JSR 248, 2006).

### 3.4 JAVA LIBRARIES FOR SIP PROGRAMMING

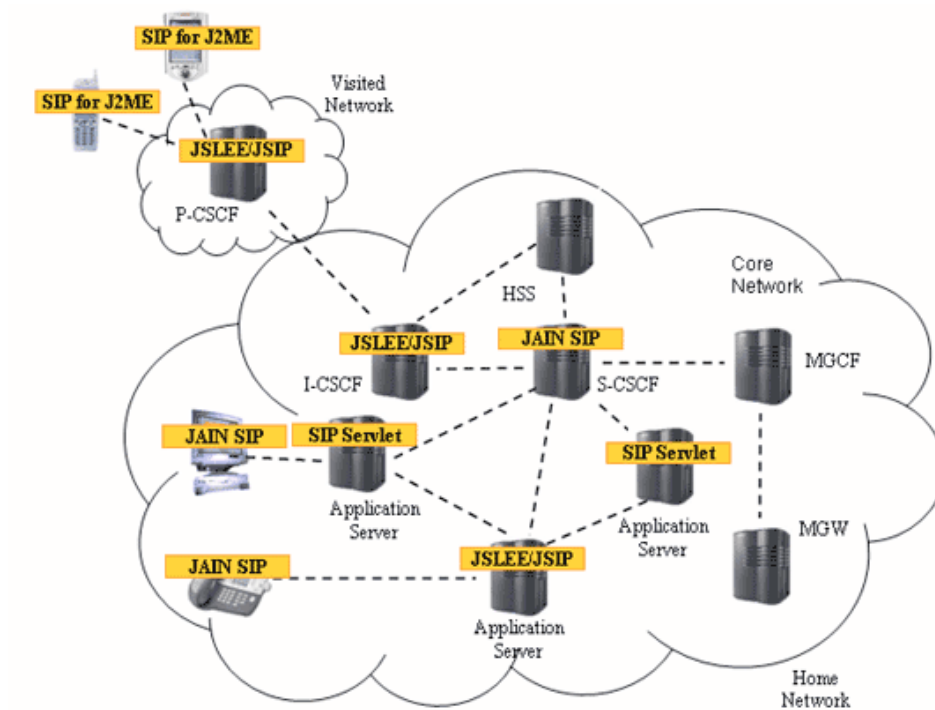


Figure 3.3: IMS and the applicability of the SIP specifications (O'Doherty, 2003)

The SIP APIs being standardized for the Java platform may be viewed as overlapping in functionality with regard to the type of applications that can be implemented within each specification (O'Doherty, 2003), as seen in Figure 3.3. For rapid development of services, programmers and application developers need APIs.

The selection of IETF SIP to become the core signalling protocol for the 3GPP IMS has resulted in the extensive use of the SIP APIs as the interface towards IMS. SIP API for J2ME has been extensively used in the development of the IMS Client for this project, therefore is briefly described in the following subsection.

#### 3.4.1 SIP API for J2ME

Nokia Corporation is the specification lead within the JCP in the development of the SIP API for J2ME, which was given the JSR number 180 (JCP JSR 180, 2004). JSR 180 API was

designed to be a compact and generic SIP API, which provides SIP functionality in transaction level. The API is integrated into the Generic Connection Framework (GCF) defined in CLDC. It is designed as an optional package that can be used with many J2ME Profiles. The minimum platform required by this API is the J2ME CLDC v1.0. The API can also be used with the J2ME CDC (JCP JSR 180, 2004).

The JSR 180 API inherited the Connector and Connection Java interfaces, as shown on the class diagram of the API in Figure 3.4.

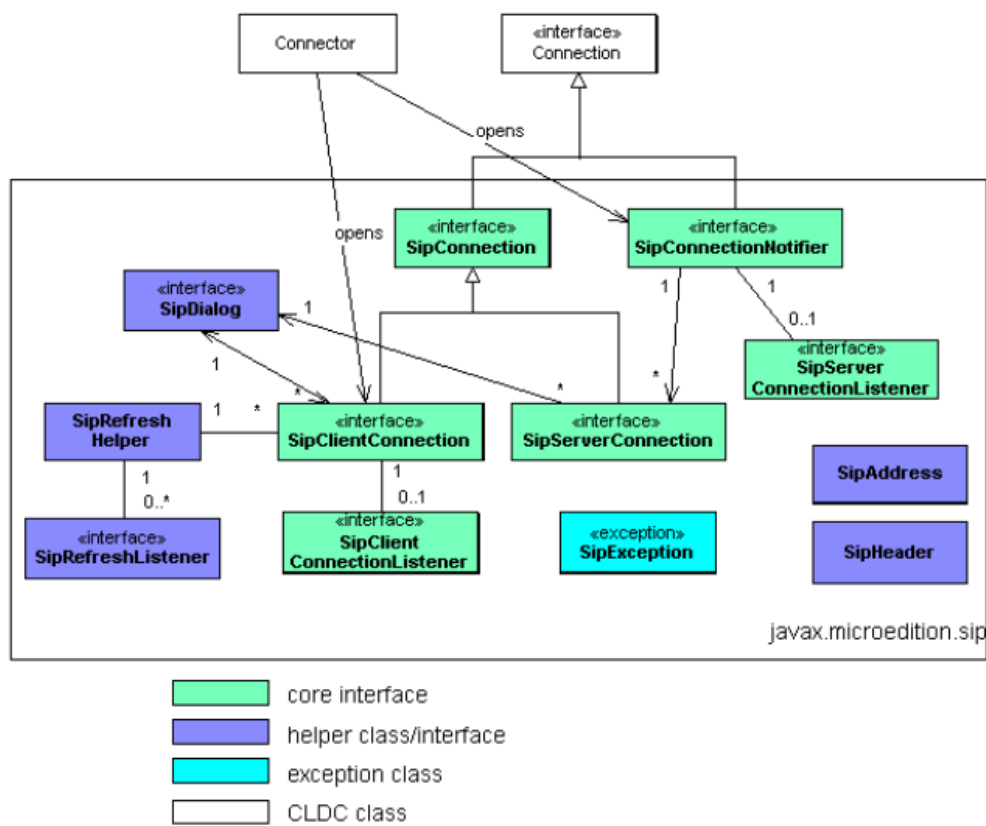


Figure 3.4: Simplified class diagram of SIP API for J2ME (JCP JSR 180, 2004)

The SIP API is used by applications to implement SIP UA functionality. This includes both UAC and UAS functionalities. A single SIP UA uses both the client and server connection functionality for SIP signalling and communication.

*i) Javax.microedition.sip Package*

This is the package defined by the JSR 180 API. It is also the main package used to implement this project's IMS Client. In this subsection, some of the major components of this package will be defined, but the full specification of the package can be found on the Java Community Process (JCP) website.

- *SipConnection* - this is the base interface for SIP connections. It holds the common properties and methods for the following sub-interfaces: *SipClientConnection* and *SipServerConnection*
- *SipClientConnection* – this interface represents SIP client transaction. Applications can create a new *SipClientConnection* with *Connector* or *SipDialog* object. This interface is also used to initialise and send responses.
- *SipSeverConnection* – this interface represents SIP server transaction. The transaction is created by the *SipConnectionNotifier* when a new request is received.
- *SipConnectionNotifier* – this interface defines a SIP server connection notifier, which is opened with *Connector.open()* using a SIP URI string with the host and user omitted. This interface queues the received messages. To receive incoming requests, application calls the *acceptAndOpen()* method, which returns a *SipServerConnection* instance.
- *SipClientConnectionListener* - this is the listener interface for incoming SIP responses.
- *SipServerConnectionListener* – this is the listener interface for incoming SIP requests.

As shown in Figure 3.5, the process of sending a message is twofold. Firstly the message is the prepared and sent. Secondly, the response is processed.

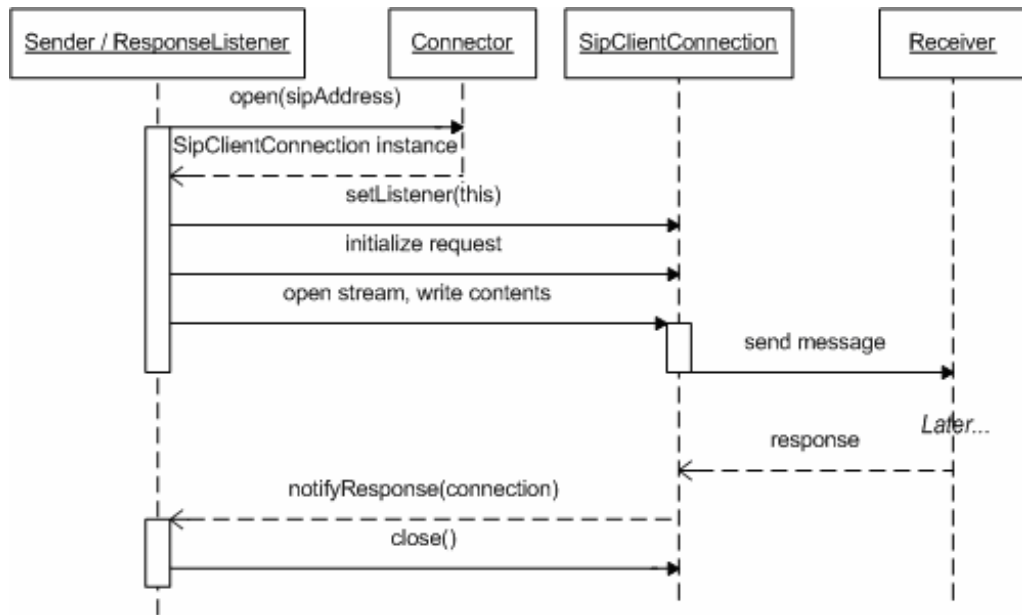


Figure 3.5: Sending SIP request (JCP JSR 180, 2004)

There are two ways to process an incoming request using the JSR 180 API. The first method uses the synchronous approach which involves blocking the current thread to wait for one to arrive. This method works well if it is known that the request will be received at a certain time. The second method is to open a permanent server connection and be notified when a message arrives asynchronously. This is sometimes referred to as the call-back loop, as shown in Figure 3.6.

Due to the flexibility of SIP, the javax.microedition.sip package was extended in this project to allow other IMS based headers and methods to be exchanged. As indicated in Chapter 2, 3GPP and IETF collaborated to ensure that the core SIP (RFC 3261) is extended to comply with the needs of the IMS. Such SIP extensions were released in the form of RFC documents, and by making use of these RFCs, the traditional SIP API was enhanced to perform most of the IMS session control.

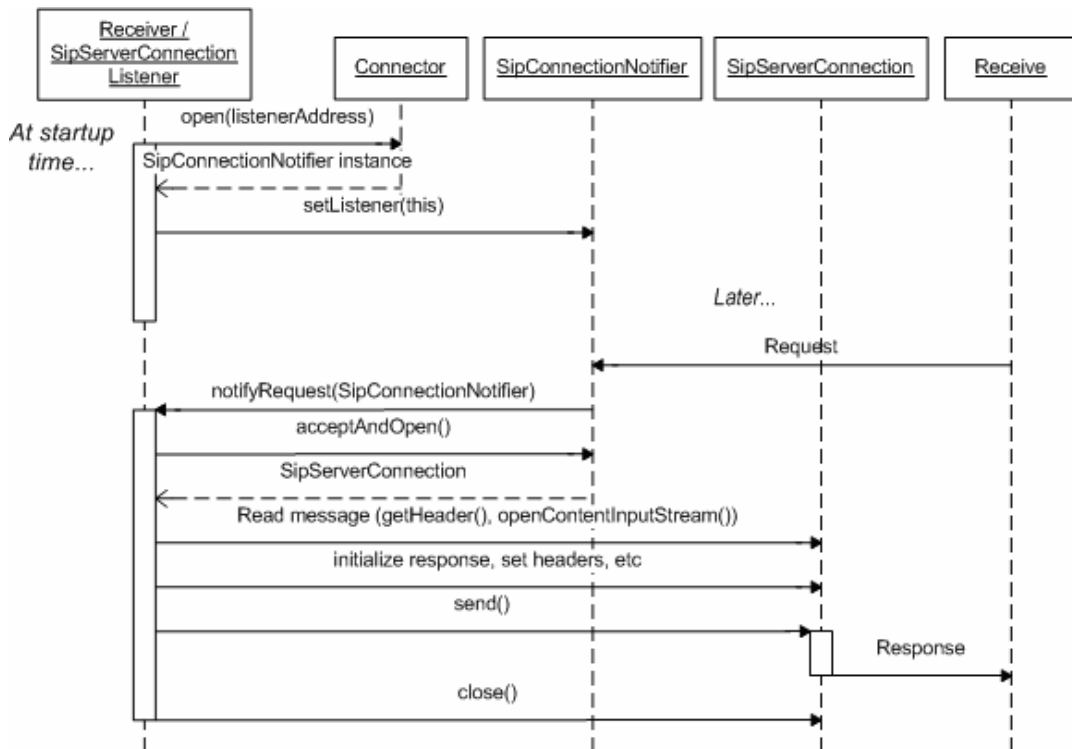


Figure 3.6: Receiving a SIP request (JCP JSR 180, 2004)

### 3.5 CONCLUSION

In this chapter, the literature review of the Java platforms and specifically J2ME platform was presented. JCP has created several APIs for quick and effective development of Java-based applications. In the next chapter, the development of the IMS Client will be presented.

## CHAPTER 4

### SYSTEM DEVELOPMENT AND IMPLEMENTATION

#### 4.1 INTRODUCTION

This chapter presents the system development and implementation based on J2ME. J2ME and SIP APIs were discussed in Chapter 3. A complete system design is presented in section 4.2 while section 4.3 presents the system implementation and simulation.

#### 4.2 SYSTEM DEVELOPMENT

In this section a detailed design and development of the IMS Client are explained. The development is based on the tools and libraries which were featured in Chapter 3. A complete IMS Client is made up of the User Agent Client (UAC) side and the User Agent Server (UAS) side, and its stack is shown in Figure 4.1. While UAC side generates the requests and processes the responses, the UAS side processes the requests and then generates the responses.

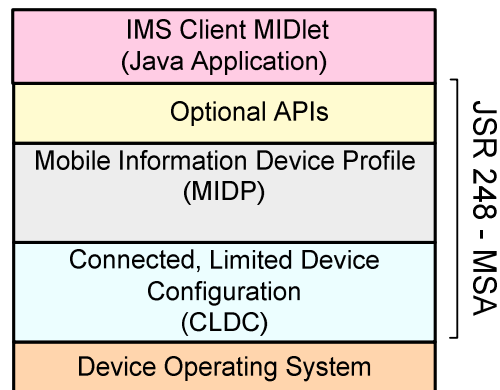


Figure 4.1: IMS Client Stack (Based on Li & Knudsen, 2005:2)

Depicted in Figure 4.2 is the position of a proposed IMS Client on the existing IMS core network and the access network. Access network can be any IPv4-connected network such as mobile 3G network or wireline Asynchronous Digital Subscriber Loop (ADSL) link.

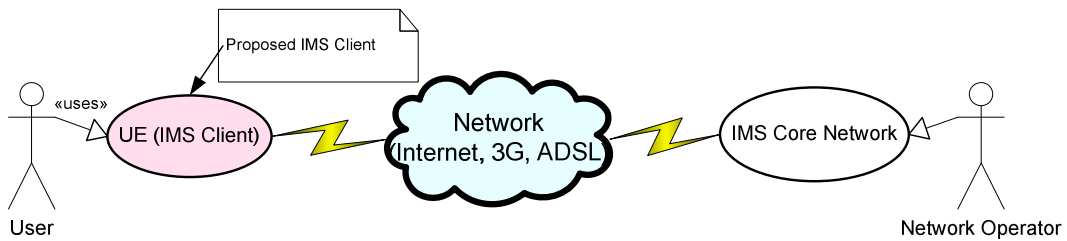


Figure 4.2: IMS Client position in relation to IMS CN

The main flow diagram of the UAC side of IMS Client is shown in Figure 4.3. The diagram shows the flow of procedures and events to be performed by the IMS Client after a successful registration, the creation of the session and the ending of the session.

The main flow diagram of the UAC side of IMS Client is shown in Figure 4.3. The diagram shows the flow of procedures and events to be performed by the IMS Client after a successful registration, the creation of the session and the ending of the session.



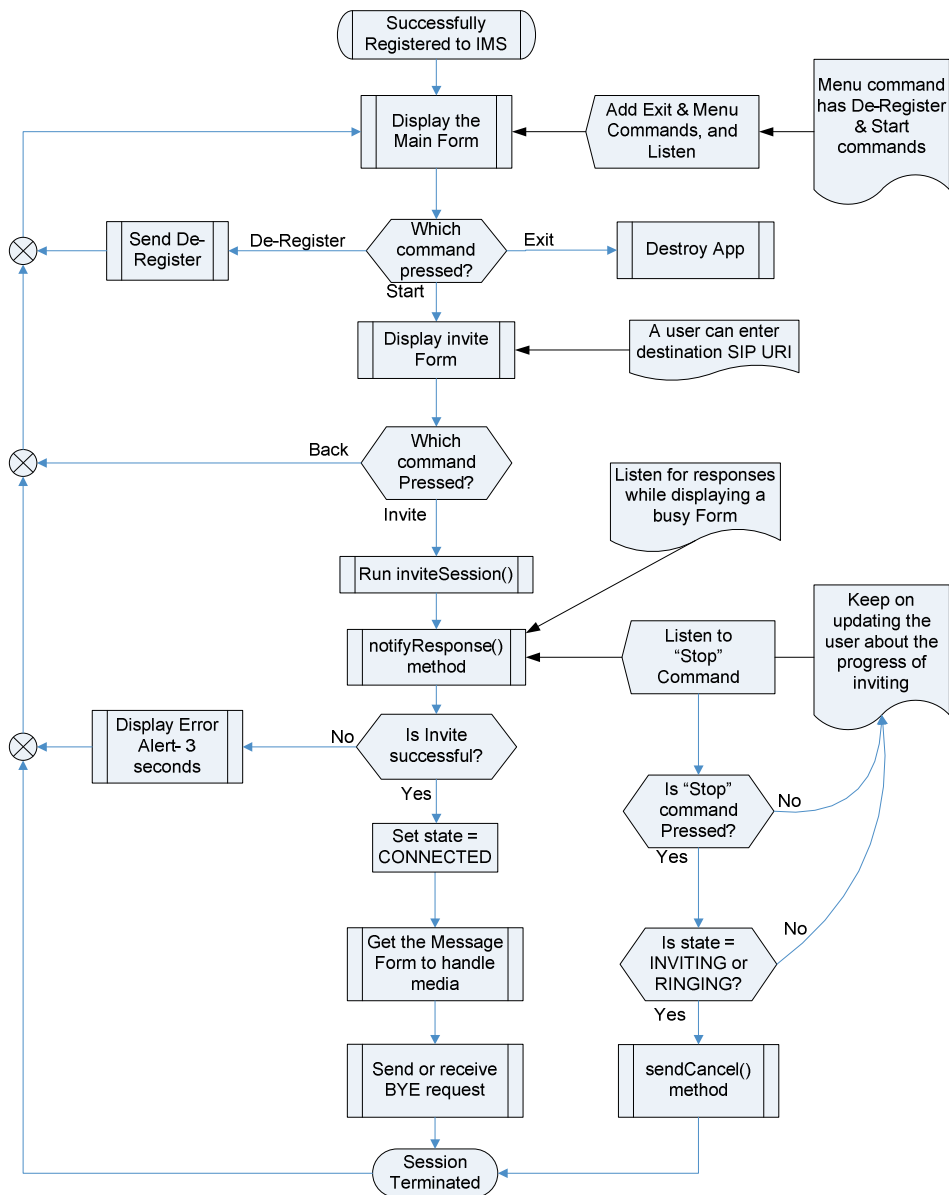


Figure 4.3: Session Establishment Main Flow diagram

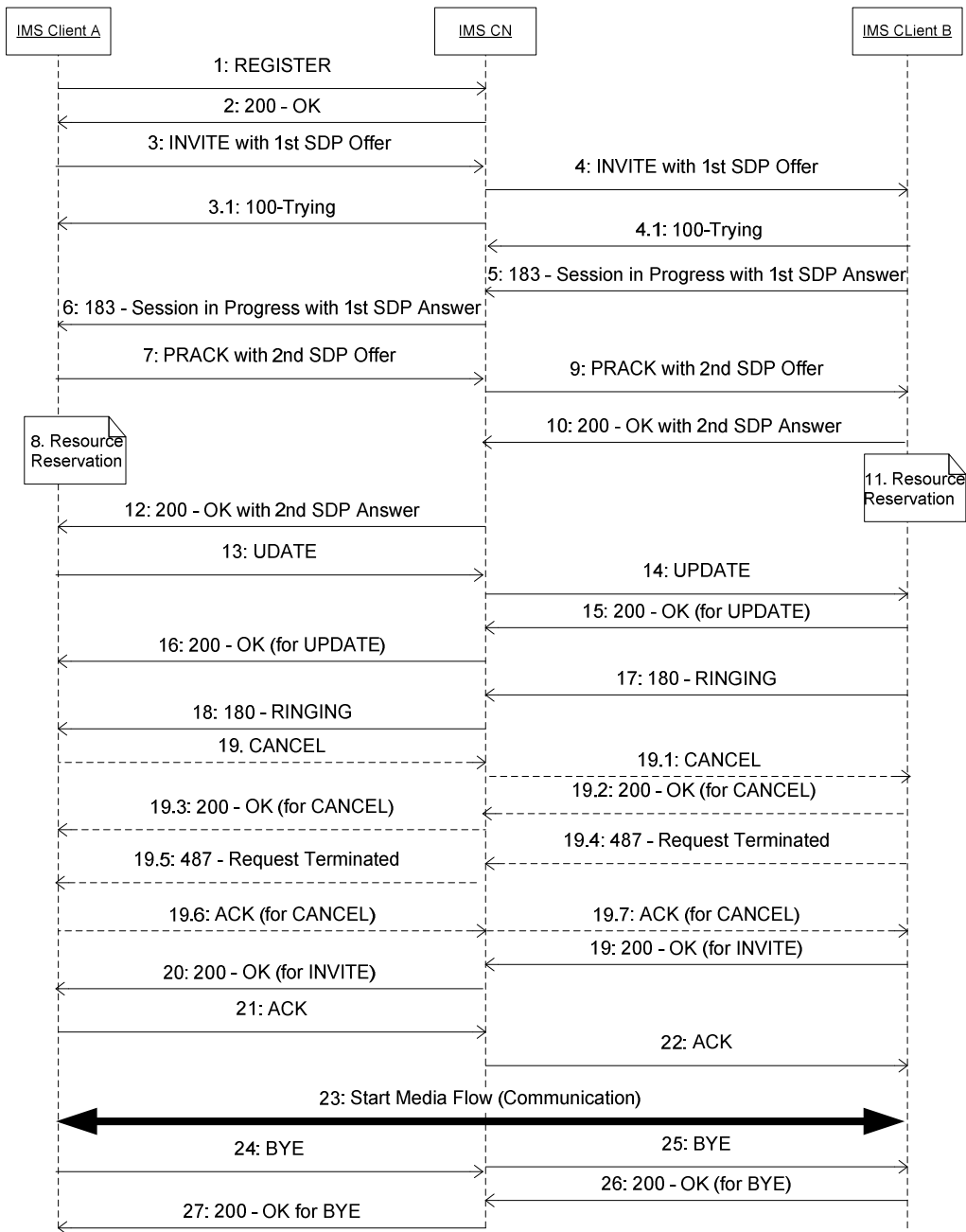


Figure 4.4: IMS Client Sequence Diagram (3GPP TR 24.930, 2007)

The system is designed to perform the following functions, as shown in Figure 4.4:

- Registering on the IMS Core Network
- Establishing a session using INVITE method

- Negotiating media codex and QoS through SDP offer/answer mechanism
- Terminating a session
- Exchanging text messages (Instant Messaging)
- Cancelling the early session using CANCEL method
- Ending the session using BYE request

#### 4.2.1 SIP Registration

As discussed in Chapter 2, each IMS subscriber needs to be registered to his/her home IMS CN to access the IMS services. Making use of the SIP API (JSR 180), a SIP REGISTER request is initiated with the important headers, and the connection is opened to the home domain registrar's URI or IP address:

```
private SipClientConnection connection = null;
s_cscf = "sip:scscf@registrar.domain:5060;transport=udp";
connection = (SipClientConnection) Connector.open(s_cscf);
connection.initRequest("REGISTER", null);
connection.setHeader("Route", p_cscfAddress);
connection.setHeader("Require", "sec-agree");
.
connection.send();
```

Due to the configuration of the IMS testbed at INT lab, a single REGISTER request is adequate to perform registration. Authentication mechanisms have not yet been implemented within this testbed. Before a user registers to the home IMS CN (S-CSCF), a user account must be created in the IMS HSS database (i.e. the allocation of public and private user identities). If the user is not known to the domain, such a user will be unable to register. After having sent the REGISTER request, the client has to wait for the final response. A 200-OK response means the user has registered successfully. A sequence diagram of the registration procedure is shown in Appendix A.1.

Figure 4.5 depicts the IMS Client display before the user can register to the home IMS CN. The IMS Client is configured to first prompt the user to register. The SIP REGISTER request is the

sent to the home IMS registrar, once the “Register” command has been pressed. If the user presses the “Exit” command, the IMS Client application will be closed and destroyed from the Sun WTK smart-phone emulator.



Figure 4.5: IMS Client Display before Registration

The Java code to create the above display is provided in Appendix B.

In order to avoid multiple registrations, the IMS Client displays a progress screen, as shown in Figure 4.6. The Client displays a “Please wait” message while it communicates with the IMS network entities. Though Figure 4.6 shows a busy display during registration, the same display is used when establishing the session.

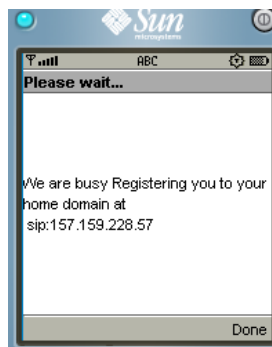


Figure 4.6: IMS Client Busy Display

If the IMS registrar or the whole IMS CN is not up and running, the client will timeout after 30 seconds, and allow the user to try again.

## 4.2.2 Session Establishment

After having successfully registered to the home IMS network (i.e. after receiving a 200-OK response), the IMS Client will display a different screen, to allow the user to establish a session or to de-register from the IMS network. The sequence diagram for session establishment is shown in Appendix A.2. Figure 4.7 depicts the IMS Client display after a successful registration to IMS network.

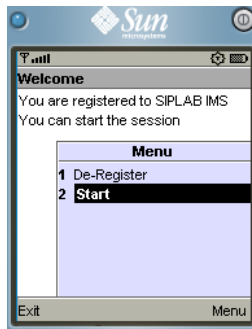


Figure 4.7: IMS Client Display after Registration

Again, if the user decides to close the IMS Client MIDlet, he presses the “Exit” command. The “Menu” command consists of two subcommands: the first one, “De-Register”, allows the user to de-register from the IMS network, while the second one; “Start” command moves the display into the invite screen, shown in Figure 4.8.

If the user follows the “De-Register” command, a SIP REGISTER request will be sent to the IMS CN. The SIP REGISTER request that is sent for de-registration is similar to the one sent for registration, with the only exception being the value of the Expires header, which must be set to zero.



Figure 4.8: IMS Client Invite Display

From the invite display, the user can establish a session by entering the destination SIP URI, and pressing an “Invite” command or return to the main display by pressing the “Back” command. When an “Invite” command is pressed, a SIP INVITE request is constructed and sent to the destination SIP URI, through the IMS network elements. IPv4 was used for the communication and registration in this study.

A captured SIP trace for an INVITE request with SDP offer that was sent from IMS Client A (the user is Moshe) to IMS Client B (the user is Vincent) is shown in Table 4.1.

Table 4.1: INVITE Request Trace

<pre> INVITE sip:vincent@157.159.103.57:5070 SIP/2.0 Call-ID: edb74ffbf1cad18788cb6339b906ec7@157.159.229.253 CSeq: 1 INVITE Via: SIP/2.0/UDP 157.xxx.xxx.xxx;branch=z9hG4bK2ea5.2b263c91.0;i=8 Via: SIP/2.0/TCP 157.xxx.xxx.xxx;branch=0;i=e Via: SIP/2.0/TCP 157.159.xxx.xxx;branch=0 Via:SIP/2.0/UDP 157.159.xxx.xxx:5070;branch=z9hG4bKdd69421d967dd7febbc60d55a43b0587 Max-Forwards: 66 P-Preferred-Identity: Moshe&lt;sip:moshe@siplab.int-evry.fr&gt; P-Called-Party-ID: &lt;sip:vincent@siplab.int-evry.fr&gt; P-Visited-Network-ID: home P-Access-Network-Info: siplab </pre>
--

```

From: Moshe <sip:moshe@siplab.int-evry.fr>;tag=1426653256
To: <sip:vincent@siplab.int-evry.fr>
Require: 100rel ,preconditions ,sec-agree
Supported: 100rel
Contact: <sip:moshe@157.159.229.253:5070>
Allow: INVITE,ACK,CANCEL,BYE,PRACK,UPDATE,REFER,MESSAGE
Content-Type: application/sdp
Content-Length: 230

v=0
o=Moshe 10022007 10022007 IN IP4 157.xxx.xxx.xxx
s=-
c=IN IP4 157.xxx.xxx.xxx
t=0 0
m=message 3150 SIP/UDP rfc3428
a=curr:qos local none
a=curr:qos remote none
a=des:qos mandatory local sendrecv
a=des:qos none remote sendrecv

```

#### 4.2.3 Session Description Protocol (SDP) Offer/Answer

When establishing the session, i.e. sending an INVITE request, the SDP (Handley, Jacobson & Perkins 2006) first offer is also sent. This SDP (Rosenberg & Schulzrinne, 2002) offer is attached to the INVITE request. The presence of the SDP payload on an INVITE request is indicated on the “Content-Type” header. The Content-Type header must contain the “sdp/application” value.

Because the IMS Client currently supports the Instant Messaging (Campbell, Rosenberg & Schulzrinne, 2002) session, the SDP offer contains the “text/plain” as the media type, as shown in this Java code:

```

HostSdp=
    "v=0\n"+
    "o=Moshe 23102007 23102007 IN IP4 " + myIP + "\n"+

```

```

"s=-\n"+
"c=IN IP4 " + myIP + "\n"+
"t=0 0\n"+
"m=message 9090 SIP/UDP rfc3428\n"+
"a=curr:qos local none\n" +
"a=curr:qos remote none\n" +
"a=des:qos mandatory local sendrecv\n"+
"a=des:qos none remote sendrecv\n";

```

As shown in Appendix A.2, SDP offer/answer is performed from the first INVITE request until the last 200-OK response of the UPDATE request.

#### 4.2.4 Communication Mode

The IMS Client developed in this study focuses on session control and session management. To demonstrate the use of the client, communication is achieved through Instant Messaging (IM). The IETF has released an RFC 3428 (Campbell *et al*, 2002), which is an extension of the core SIP (RFC 3261) that allows the transfer of Instant Messages. In this RFC 3428, a MESSAGE method is defined.

A Java snapshot for sending an IM is shown below:

```

1) public void sendMessage() {
2)     SipClientConnection client_connection = null;
3)     OutputStream output = null;
4)     client_connection = dialog.getNewClientConnection("MESSAGE");
5)     client_connection.setHeader("Content-Type", "text/plain");
6)     client_connection.setHeader("Content-Length",
Integer.toString(message.length()));
7)     output = client_connection.openContentOutputStream();
8)     output.write(message.getBytes());
9)     output.close();
10) }

```



The SIP MESSAGE method is sent within an existing dialog as shown on line 4 of the Java code above. When sending a request within an existing dialog, “getNewClientConnetion()” method is called with an intended method as a parameter.

#### 4.2.5 Cancelling an Early Session

Cancelling a SIP session is performed using a SIP CANCEL method as specified in RFC 3261 (Rosenberg *et al*, 2002). An IMS Client user can cancel a request by pressing a “Stop” command during session establishment. A session will be cancelled only if the final 200-OK for an INVITE request has not been sent yet.

#### 4.2.6 Ending Session

A session can be ended by either user pressing an “End” command. When a button is pressed, a UAC side of an IMS Client will process and send a BYE request. Based on RFC 3261 (Rosenberg *et al*, 2002), an IMS Client may not send a BYE request outside of a dialog. The caller’s IMS Client may send a BYE request for either confirmed or early dialogs and the callee’s IMS Client may send a BYE request on confirmed dialogs, but may not send a BYE request on early dialogs.

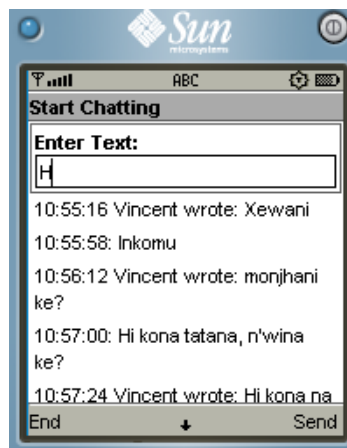


Figure 4.9: An IMS Client Chatting Display

Figure 4.9 depicts the commands appearing on an IMS Client display during IM communication. A “Send” command is used to send the typed text, while an “End” command is used to send a BYE request. Once the UAC on an IMS send a BYE request, it will wait for a 200-OK final response to ensure that a session has ended.

### 4.3 SYSTEM IMPLEMENTATION

In this section, an implementation of the IMS Client within the IMS testbed is discussed. The implementation was carried out using the development tools and libraries described in Chapter 3. Sub-section 4.3.1 discusses the IMS testbed and the implementation environment for our study, while sub-section 4.3.2 provides the system simulation.

#### 4.3.1 IMS Network Testbed

This study was carried out at the INT laboratory at Evry in France. The lab was originally built for SIP experimenting. It was then upgraded to be compliant with the standardized IMS network. As the result, work is still in progress to finalize the migration from SIP to full functional IMS architecture. Figure 4.10 shows the whole INT IMS Testbed SIP lab.

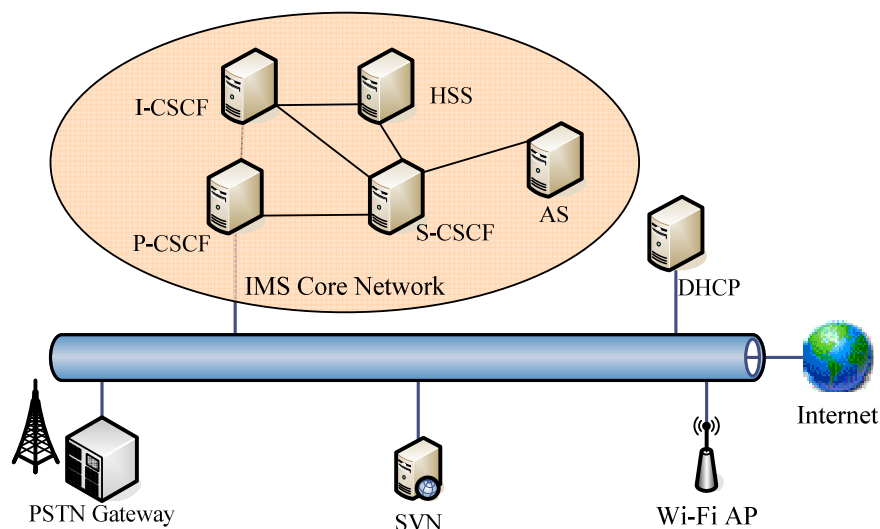


Figure 4.10: INT IMS Testbed

To perform the experiments on this testbed, the user credentials first have to be created within the HSS. The private user identity was created as: moshe.siplab.int-evry.fr. The IMS Testbed can be accessed through either the LAN or Wi-Fi technologies within the INT network infrastructure.

### **4.3.2 System Simulation**

Session setup, control and management are at the core of this study. The IMS Client uses SIP and SDP to manage and perform these actions. To ensure a functional system, Instant Messaging was used to communicate between two users. The discussion of the implementation of the IMS Client is presented in this subsection.

#### *i. Creating the IMS Client*

A project was created using Sun WTK KToolbar (shown in Figure 4.11) and it was named “Moshe\_IMS”. If the Sun WTK has been installed on the C drive, on Windows machine, the project folder, Moshe\_IMS, is created in the Sun WTK application directory as follows:

- C:\WTK25\apps\Moshe\_IMS

When creating a project using the Sun WTK, the following folders are automatically created:

- src – stores the source code
- bin – stores the JAR and Java Application Descriptor (JAD) files
- classes – stores the class files of the compiled application
- res – stores the resource files, like pictures
- lib – stores the library files

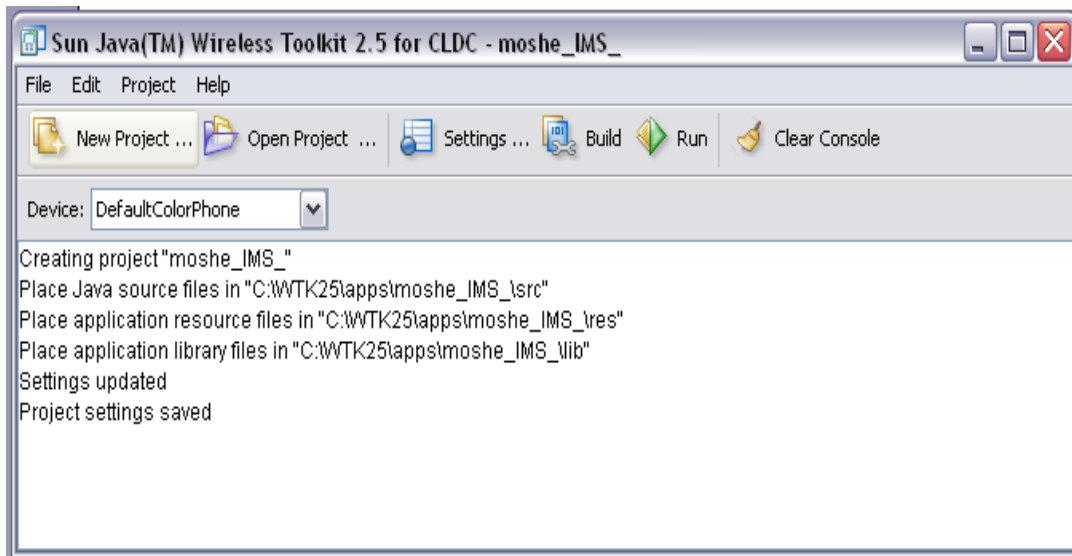


Figure 4.11: Sun Wireless Toolkit – KToolbar

The Java source code was edited in EclipseMe IDE and stored in the src folder. The source code was compiled and pre-verified by building it with the Sun WTK. After the project has been built, the compiled files are stored in the Classes folder.

The next step is to package the project. Packaging is done using the Sun WTK Package submenu (this is under Project → Package → Create Package). By packaging the project, the MIDlet suite JAR and JAD files are created and stored in the bin folder. This MIDlet suite is ready to be deployed into a mobile device emulator or a real device. The JAR and JAD files are the important files used to deploy the MIDlet into the emulated or real devices.

For this study, a Nokia phone emulator, provided with the Sun WTK was used to emulate the real mobile device. Sun WTK version 2.5 supports the MSA features, which include the SIP API for J2ME.

*ii. Running the IMS Client*

The “Run” command on the Sun WTK KToolbar is used to run the MIDlet. When the MIDlet is started, the registration screen is displayed. As stated before, registration is the first step towards accessing IMS services. After a successful registration to the home IMS network, an

invite screen, which allows the user to establish a session is displayed. After a successful invitation, the chatting screen is displayed. The chatting screen allows the user to type text and send to the other remote user. Communication can be ended by pressing the “End” command on the chatting screen.

#### **4.4 CONCLUSION**

This chapter detailed the development tools, libraries and software for the Light-Weight IMS Client. The J2ME platform is the main tool used for the development of CLDC and MIDP handset applications. SIP API for J2ME is important to ensure the usage of SIP protocol within the J2ME platform. The system design and implementation were also described in this chapter. Chapter 5, which is the next chapter, presents the experiment conducted in testing the system, plus the results.

## CHAPTER 5

### SYSTEM EXPERIMENTS AND RESULTS

#### 5.1 INTRODUCTION

This chapter presents the system experiments conducted and the results based on the design that was described in Chapter 4. The experimental setup and requirements in terms of software and hardware are outlined. The experiment was performed using two Light-Weight IMS Clients within the IMS Testbed that is available at the INT-Evry laboratory. The evaluation and discussion of the results are also included in this chapter.

#### 5.2 EXPERIMENTAL SETUP

The Light-Weight IMS Client was tested in an experimental setup within the INT lab. The experimental setup is shown on Figure 5.1. The IMS Testbed comprises the IMS CN, which provides the basic control layer elements such as: P-CSCF, I-CSCF, S-CSCF and HSS. The Dynamic Host Configuration Protocol (DHCP) and the Application Server (AS) also form part of the IMS Testbed. Each IMS network element has its own IPv4 address. The VMware software is used to interact, from the same screen, with the P-CSCF, I-CSCF and S-CSCF, where each network element can be restarted or stopped. The HSS uses a separate machine, and with its own IP address and display.

The experiment was conducted using two identical IMS Clients, hosted from two different machines, but in the same Local Area Network (LAN), as shown in Figure 5.1. From the IMS Testbed, two public and private user identities were created, one for Moshe and the other one for Vincent.

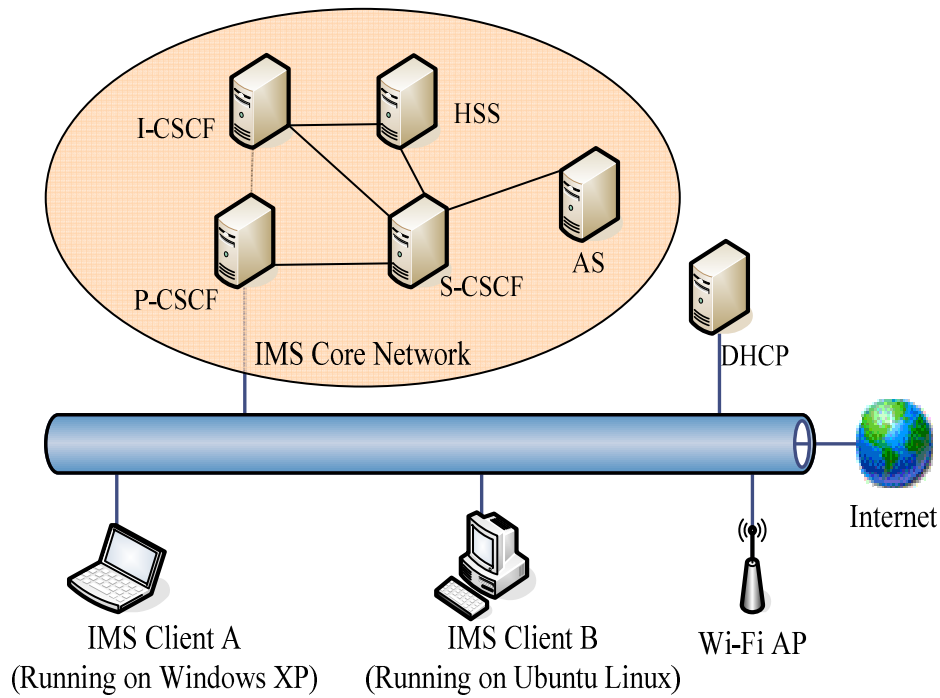


Figure 5.1: Experimental Setup at INT IMS Testbed

### 5.3 TESTING REQUIREMENTS

The list of hardware and software requirements, and tools used for setting up the experiment are provided in the following subsections. The aim of this project was not to develop the IMS testbed; hence only the specifications for this testbed as outlined in subsection 5.3.3 were provided.

#### 5.3.1 Hardware Requirements

Table 5.1 lists the hardware requirements for conducting the experiment using two IMS Clients and an IMS network (testbed).

Table 5.1: Hardware Specifications

Name	Description
<b>Personal Computer (PC)</b>	
CPU	Intel Pentium 4, 2.66 GHz

RAM	512 MB
Hard Drive	80 GB
Network Connection	Intel PRO/100 VE
<b>Laptop</b>	
CPU	Intel Centrino Duo 1.66 GHz
RAM	1 GB RAM
Hard Drive	80 GB
Network Connection	Intel PRO/100 VE and Intel PRO/Wireless 3945ABG

### 5.3.2 Software Requirements

Table 5.2 lists the software requirements for conducting the experiment using two IMS Clients and an IMS network (testbed).

Table 5.2: Software Specifications

Name	Description
<b>Personal Computer (PC)</b>	
Operating System	Linux Ubuntu 6
Java	Jdk 1.5.1, Java Virtual Machine(JVM)
J2ME platform	Linux version of Sun Wireless Toolkit 2.5.2
EclipseME	Source code editor. A plug-in to Eclipse
Phone Emulator	Provided within Sun WTK
<b>Laptop</b>	
Operating System	Microsoft Windows XP Home Edition
Java	Jdk 1.5.0_09 JVM
J2ME platform	Windows version of Sun WTK 2.5.1
EclipseME	A source code editor. A plug-in to Eclipse 3.2
Phone Emulator	Provided within Sun WTK
<b>Other Software</b>	
Wireshark	A network protocol analyzer used to capture network packets (formally known as Ethereal).



### 5.3.3 INT IMS Testbed Specifications

The INT IMS testbed is a work-in-progress project, being carried out within the multimedia department at INT-Evry labs. The specifications provided in this subsection were provided by this team of students who were responsible for building the testbed.

Table 5.3: INT IMS Testbed Specifications

Name	Description
<b>Hosts</b>	
CPU	Pentium 4 @3.40GHz
RAM	2.0 GB
Server	Windows Server 2003 Enterprise Edition
	VMWare Server 1.0.2
<b>Guests</b>	
CPU	Shared CPU
RAM	256 MB
GNU/Linux Debian	Core 2.6.8 arch i686
SIP	SIP Express Router 0.8.14
Database	MySQL server 5.0.20
<b>Network</b>	
Server	Windows DHCP server for clients
Router	Cisco 400 Series (10 Mbps interfaces)
Switch	MRV MR2228-S2C (100 Mbps Interfaces)
Infrastructure Type	Bus Type

### 5.4 EXPERIMENTAL DEMONSTRATION

The IMS Client was tested on the setup shown in Figure 5.1. Both Clients first had to register to the IMS network. After registration, IMS Client A established a session by sending an

INVITE request to IMS Client B. Figure 5.2 shows the testing environment at INT lab, from the IMS testbed.

The P-CSCF, I-CSCF and the S-CSCF are hosted within the same machine, while the HSS is on a separate machine. The IMS Client application or MIDlet was deployed within two separate computers. IMS Client A was deployed within the laptop, while IMS Client B was deployed within the personal computer.

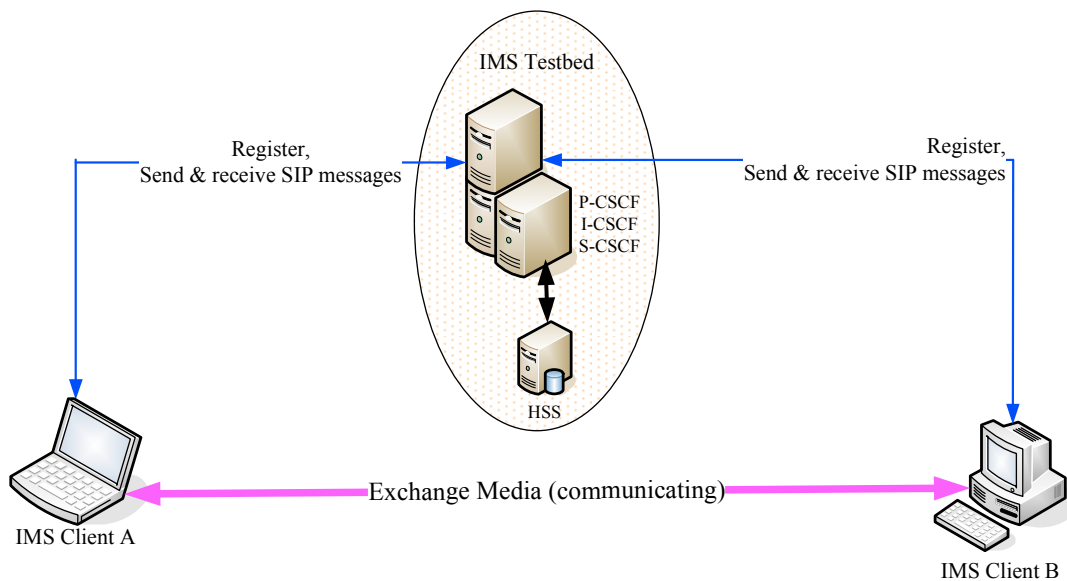


Figure 5.2: INT IMS Testing Environment

During registration and session establishment, the signalling path was used. All the signalling messages traversed the IMS network, while the payload media used a different channel. Media was exchanged directly between the two IMS Clients, without passing through the IMS network.

Figure 5.3 shows one of the IMS Client screenshots from the start until the end of the session. At the beginning, the user is given a screen with an option to register to the IMS network. After having registered successfully, the user can decide to establish the session or to de-register from the home IMS network. To get the invite screen, the users use the “Start” command. Within the invite screen, a user is requested to enter the destination SIP address, and then press an “Invite” command to send a SIP INVITE request.

An INVITE request will follow the same path used during registration (i.e. go through the home IMS network towards the destination user). The home IMS network will forward an INVITE request to the destination user provided such a user is registered. If the user is not registered, the IMS network will return the final response to the originating user, indicating that the other user is not available.

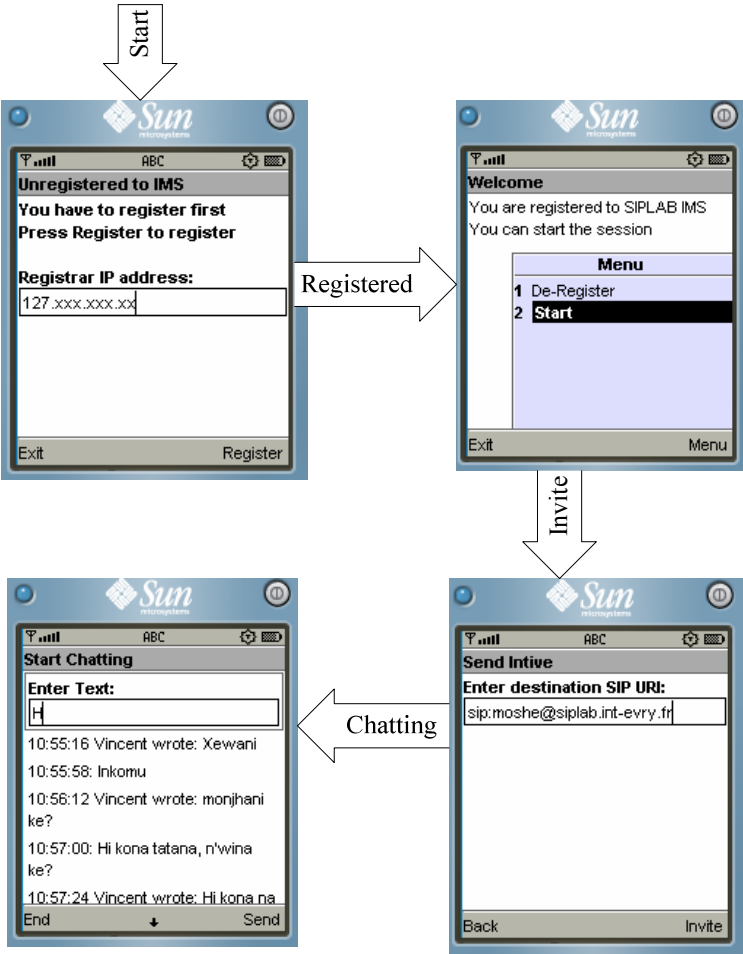


Figure 5.3: Demonstration of How to Establish a Session

**5.5 RESULTS DISCUSSION**

In this section the evaluation and discussion of the results based on the experimental demonstration carried out on the previous section will be done. In the next section, the results as extracted from the Wireshark® Internet protocol analyser will be discussed. The aim is to

show and discuss the SIP messages (requests and responses) exchanged between the two IMS Clients and the IMS testbed. The SDP content will also form the basis of the results discussion.

Poikselkä *et al* (2004:76) indicated that one of the most important issues of the IMS design was to separate the control plane from the user plane. But there was a need to find the mechanism to authorize and control the use of the bearer traffic intended for IMS media traffic. This mechanism that was found is based on SDP parameters negotiated at the IMS session.

### 5.5.1 Registration Request

The importance of registration was discussed in the previous chapters (Chapter 2 and Chapter 3). Figure 5.4 shows the SIP REGISTER request sent from Vincent IMS Client to the home IMS network (which is the IMS testbed). The registering IMS Client's SIP URI is placed under the "To:" and "From:" headers. The "To:" header does not have any tag. Because this is the original request, only the "From:" header can have a tag number.

No. -	Time	Source	Destination	Protocol	Info
166	28.099767	157.159.103.57	157.159.103.29	SIP	Request: REGISTER
167	28.150073	157.159.103.29	157.159.103.57	SIP	Status: 200 OK

⊕ User Datagram Protocol, Src Port: 1963 (1963), Dst Port: 5060 (5060)					
⊖ Session Initiation Protocol					
⊕ Request-Line: REGISTER sip:157.159.103.29:5060 SIP/2.0					
⊖ Message Header					
Call-ID: 6821900b0f3a30d9e16a66ec1722f426@157.159.103.57					
⊕ CSeq: 1 REGISTER					
⊕ Via: SIP/2.0/UDP 157.159.103.57:5060;branch=z9hg4bk74eedee6f53222af553640798bf2d007					
Max-Forwards: 70					
Require: sec-agree					
Supported: path					
⊕ Contact: <sip:vincent@157.159.103.57:5070>;expires=3600					
Allow: INVITE, ACK, CANCEL, BYE, PRACK, UPDATE, REFER, MESSAGE					
⊕ From: Vincent <sip:vincent@siplab.int-evry.fr>;tag=46969796					
⊕ To: Vincent <sip:vincent@siplab.int-evry.fr>					
User-Agent: Sun WTK 2.5 Emulator					
Private: none					
Content-Length: 0					

Figure 5.4: REGISTER Request Trace

The Contact header that is very important, is mandatory in the SIP REGISTER request. It must include the host device's IP address, port number and the expiry period. The IMS Client will be registered to the home IMS once it has received a 200-OK response.

### 5.5.2 Invite Request

Figure 5.5 depicts the SIP INVITE request, originated by Vincent IMS Client and sent to Moshe IMS Client. The INVITE request contains the SIP headers and the SDP message. The QoS in IMS is guaranteed during a SIP session setup or session modification procedure when an IMS Client negotiates its capabilities and expresses its QoS requirements (Poikselkä *et al*, 2006:13). The attribute lines or a-lines under the m-line of the SDP form a block that describes in more detail the media of the m-line under which they are placed. To understand these SDP attribute tags, a brief explanation is given in Table 5.4.

Table 5.4: SDP Attributes

Attribute Tags	Description
Curr (Current)	indicates the extent to which the set preconditions have already been fulfilled.
Des (Desired)	makes it possible to set desired preconditions for the local and the remote user.
Qos (Quality of Service)	The precondition is set due to certain quality requirements of the related media stream.
Local	The local user, who originates the session.
Remote	indicates the destination.
None	indicates that no parameter was set
Mandatory	Indicates a mandatory precondition.
Sendrecv (Send and Receive)	tools should start in send and receive mode

The IETF has released a RFC 4566 document which supports the following media types. These are: "audio", "video", "text", "application", and "message". This list may be extended in

the future. In this study, the IMS Client will support only the “message” as the media type, as shown in all the SDP messages.

Note 1 in Figure 5.5 draws the attention to the desired QoS attribute from the local or calling subscriber. The local user indicates the desired (des) precondition to start a session in send and receive mode (sendrecv), as indicated by the “mandatory” tag.

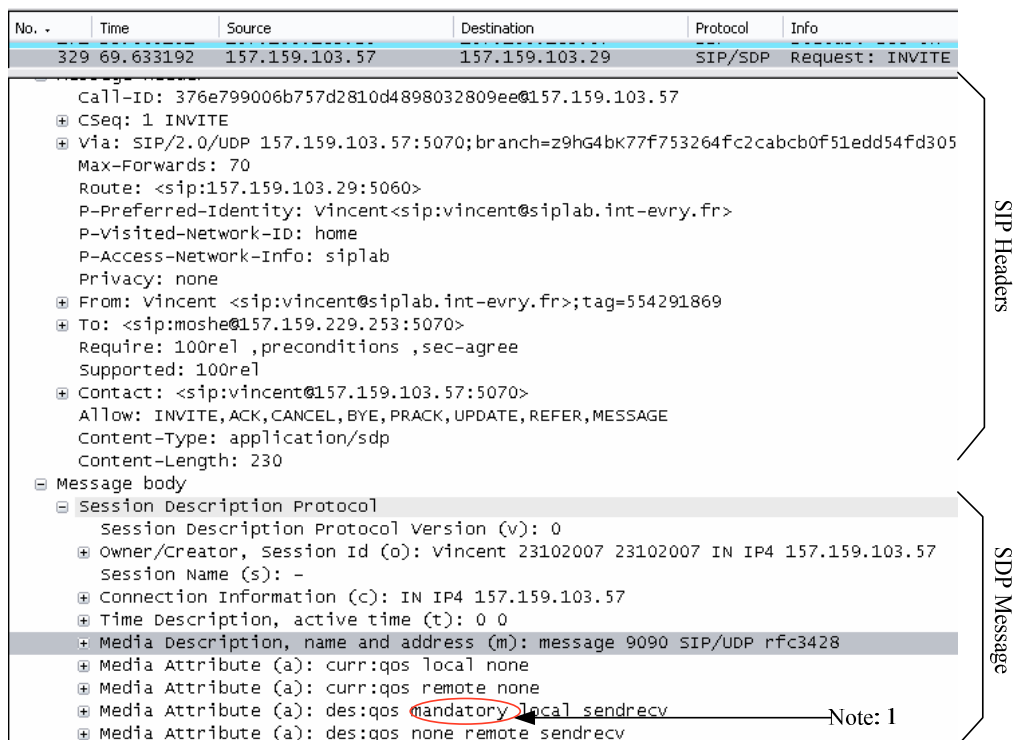


Figure 5.5: INVITE Request Trace

### 5.5.3 Trying Response

To avoid frequent re-transmission of the INVITE request from the UE, the P-CSCF sends back a 100-Trying response after it has received the INVITE, to indicate that from here on the P-CSCF will take care of the re-transmissions.

No. -	Time	Source	Destination	Protocol	Info
372	74.760735	157.159.229.253	157.159.103.29	SIP	Status: 100 Trying
373	74.771079	157.159.103.29	157.159.103.57	SIP	Status: 100 Trying
374	74.778142	157.159.229.253	157.159.103.29	SIP/SDP	Status: 183 Session

```

Session Initiation Protocol
├─ Status-Line: SIP/2.0 100 Trying
├─ Message Header
│   └─ Call-ID: 376e799006b757d2810d4898032809ee@157.159.103.57
│       └─ CSeq: 1 INVITE
│           └─ From: "vincent" <sip:vincent@siplab.int-evry.fr>;tag=554291869
│               └─ To: <sip:moshe@157.159.229.253:5070>;tag=1604172010
│                   └─ Require: 100rel,preconditions,sec-agree
│                       └─ Contact: <sip:157.159.229.253:5070;transport=udp>
│                           └─ Via: SIP/2.0/UDP 157.159.103.57:5070;branch=z9hg4bk77f753264fc2cabcb0f51edd54fd3050
│                               └─ Content-Length: 0

```

Figure 5. 6: 100-Trying Response Trace

The 100-Trying response is sent by all call-statefull SIP proxies on the route, and it is always stopped at the SIP proxy that is the last to take over responsibility for re-transmission, as shown in Figure 5.6.

#### 5.5.4 Session Progress Response

The 183-Session-in-Progress response is sent by the remote client to indicate that session establishment procedures have been started, but the called user has not been informed yet. It is sent immediately on receipt of the initial INVITE request. It also contains the SDP first answer, as shown in Figure 5.7. The remote user will indicate the desire to start a session in the send and receive mode as indicated by Note 2 in Figure 5.7. Going back to Figure 5.5, the remote sendrecv tag was set to “none”. And now it is set to “mandatory”. The media type is also set to “message”, but with different port number. This means that both IMS Clients support the same, single media type, which is “message”.

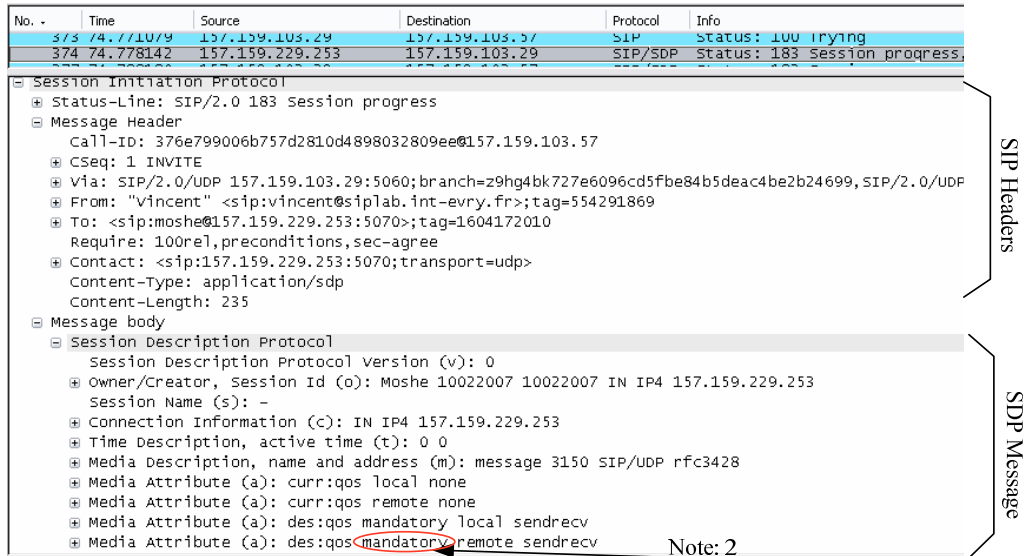


Figure 5.7: 183-Session Progress Response Trace

### 5.5.5 PRACK Request

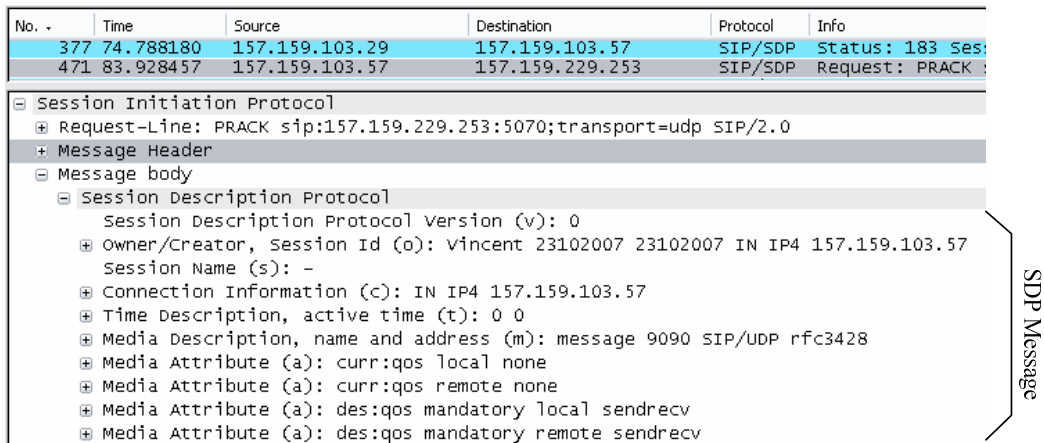


Figure 5.8: PRACK Request Trace

The local client sends back a PRACK request to acknowledge the received 183-Session in Progress response. This PRACK also carries the second SDP offer. The final decision is on media is done in this request. Because both IMS Clients involved in this session support the same media type, a decision is made and the PRACK is sent to confirm the final decision with the remote client, as shown in Figure 5.8.



The remote client responds to the PRACK request with a 200-OK response as shown in Figure 5.9.

No. -	Time	Source	Destination	Protocol	Info
573	99.113752	157.159.229.253	157.159.103.57	SIP/SDP	Status: 200 OK,
596	103.239122	157.159.103.57	157.159.229.253	SIP/SDP	Request: UPDATE

```

Session Initiation Protocol
├─ Status-Line: SIP/2.0 200 OK
├─ Message Header
│   └─ Call-ID: 376e799006b757d2810d4898032809ee@157.159.103.57
│   └─ CSeq: 2 PRACK
│   └─ Via: SIP/2.0/UDP 157.159.103.57:5070;branch=z9hG4bK244d7148506a5b86869c04d90e13ab1
│   └─ From: Vincent <sip:vincent@siplab.int-evry.fr>;tag=554291869
│   └─ To: <sip:moshe@157.159.229.253:5070>;tag=1604172010
│   └─ Contact: <sip:157.159.229.253:5070;transport=udp>
│   └─ Content-Type: application/sdp
│   └─ Content-Length: 235
├─ Message body
│   └─ Session Description Protocol
│       └─ Session Description Protocol Version (v): 0
│       └─ Owner/Creator, Session Id (o): Moshe 10022007 10022007 IN IP4 157.159.229.253
│       └─ Session Name (s): -
│       └─ Connection Information (c): IN IP4 157.159.229.253
│       └─ Time Description, active time (t): 0 0
│       └─ Media Description, name and address (m): message 3150 SIP/UDP rfc3428
│       └─ Media Attribute (a): curr:qos local none
│       └─ Media Attribute (a): curr:qos remote none
│       └─ Media Attribute (a): des:qos mandatory local sendrecv
│       └─ Media Attribute (a): des:qos mandatory remote sendrecv

```

SIP Headers

SDP Message

Figure 5.9: 200-OK Response for PRACK Trace

### 5.5.6 UPDATE Request

The local client sends the UPDATE request on receipt of the 200-OK (PRACK) response. Now the codec and resources have been reserved for the session; and the local user indicates its current qos to the remote client as indicated by Note 3 in Figure 5.10.

No. -	Time	Source	Destination	Protocol	Info
573	99.113752	157.159.229.253	157.159.103.57	SIP/SDP	Status: 200 OK, with session description
596	103.239122	157.159.103.57	157.159.229.253	SIP/SDP	Request: UPDATE sip:157.159.229.253:5070;tran
602	104.150305	157.159.229.253	157.159.103.57	SIP/SDP	Status: 200 OK, with session description
637	107.772205	157.159.229.253	157.159.103.57	SIP/SDP	Request: UPDATE sip:157.159.229.253:5070;tran

```

User Datagram Protocol, Src Port: 1972 (1972), Dst Port: 5070 (5070)
Session Initiation Protocol
  Request-Line: UPDATE sip:157.159.229.253:5070;transport=udp SIP/2.0
  Message Header
  Message body
    Session Description Protocol
      Session Description Protocol Version (v): 0
      Owner/Creator, Session Id (o): Vincent 23102007 23102007 IN IP4 157.159.103.57
      Session Name (s): -
      Connection Information (c): IN IP4 157.159.103.57
      Time Description, active time (t): 0 0
      Media Description, name and address (m): message 9090 SIP/UDP rfc3428
      Media Attribute (a): curr:qos local mandatory ← Note: 3
      Media Attribute (a): curr:qos remote none
      Media Attribute (a): des:qos mandatory local sendrcv
      Media Attribute (a): des:qos mandatory remote sendrcv
  
```

Figure 5.10: UPDATE Request Trace

The remote client, on receipt of the UPDATE request, will respond with a 200-OK response containing the SDP final answer. The remote client will also indicate its current (curr) qos preconditions by changing the parameter from “none” to “mandatory” as indicated by Note 4 in Figure 5.11.

No. -	Time	Source	Destination	Protocol	Info
607	110.029721	157.159.103.57	157.159.229.253	SIP/SDP	Request: UPDATE sip:157.159.229.253:5070;
683	118.819300	157.159.103.57	157.159.229.253	SIP/SDP	Request: UPDATE sip:157.159.229.253:5070;
684	119.297244	157.159.229.253	157.159.103.57	SIP/SDP	Status: 200 OK, with session description
687	119.315912	157.159.229.253	157.159.103.29	SIP	Status: 180 Ringing

```

User Datagram Protocol, Src Port: 5070 (5070), Dst Port: 5070 (5070)
Session Initiation Protocol
  Status-Line: SIP/2.0 200 OK
  Message Header
    Call-ID: 376e799006b757d2810d4898032809ee@157.159.103.57
    CSeq: 3 UPDATE
    Via: SIP/2.0/UDP 157.159.103.57:5070;branch=z9hG4bKc6ad53322df290903435d4a53eeb93b
    From: Vincent <sip:vincent@siplab.int-evry.fr>;tag=554291869
    To: <sip:moshe@157.159.229.253:5070>;tag=1604172010
    Contact: <sip:157.159.229.253:5070;transport=udp>
    Content-Type: application/sdp
    Content-Length: 245
  Message body
    Session Description Protocol
      Session Description Protocol Version (v): 0
      Owner/Creator, Session Id (o): Moshe 10022007 10022007 IN IP4 157.159.229.253
      Session Name (s): -
      Connection Information (c): IN IP4 157.159.229.253
      Time Description, active time (t): 0 0
      Media Description, name and address (m): message 3150 SIP/UDP rfc3428
      Media Attribute (a): curr:qos local mandatory
      Media Attribute (a): curr:qos remote mandatory ← Note: 4
      Media Attribute (a): des:qos mandatory local sendrcv
      Media Attribute (a): des:qos mandatory remote sendrcv
  
```

Figure 5.11: 200-OK Response for UPDATE Trace

### 5.5.7 ACK Request

The local client generates and sends an ACK request for a 200-OK INVITE response to the remote client. The header fields of the ACK are constructed in the same way as that for any

request sent within a dialog as shown in Figure 5.12. The sequence number of the CSeq header field must be the same as the INVITE being acknowledged, but the CSeq method must be ACK. The ACK contains the same credentials as the INVITE.

No. -	Time	Source	Destination	Protocol	Info
849	141.200355	157.159.103.29	157.159.103.57	SIP	Status: 200 OK
890	146.485830	157.159.103.57	157.159.229.253	SIP	Request: ACK sip:
952	160.665990	157.159.103.57	157.159.229.253	SIP	Request: MESSAGE
954	160.020000	157.159.103.57	157.159.229.253	SIP	Request: MESSAGE

```

Session Initiation Protocol
  Request-Line: ACK sip:157.159.229.253:5070;transport=udp SIP/2.0
  Message Header
    Call-ID: 376e799006b757d2810d4898032809ee@157.159.103.57
    CSeq: 1 ACK
    Via: SIP/2.0/UDP 157.159.103.57:5070;branch=z9hg4bk8238a2cb9a32f1cca196c30ac85ca96e
    Max-Forwards: 70
    P-Preferred-Identity: vincent<sip:vincent@siplab.int-evry.fr>
    P-Visited-Network-ID: home
    P-Access-Network-Info: siplab
    Privacy: none
    From: Vincent <sip:vincent@siplab.int-evry.fr>;tag=554291869
    To: <sip:moshe@157.159.229.253:5070>;tag=1604172010
    Require: 100rel ,preconditions ,sec-agree
    Supported: 100rel
    Contact: <sip:vincent@157.159.103.57:5070>
    Allow: INVITE,ACK,CANCEL,BYE,PRACK,UPDATE,REFER,MESSAGE
    Content-Length: 0
  
```

Figure 5.12: ACK Request Trace

### 5.5.8 Communication Mode

Communication between the two Light-Weight IMS Clients was achieved making use of the SIP MESSAGE method. This method carries plain text as the content type, as shown in Figure 5.13. For every message sent from one client to the other, a 200-OK response should also be sent. Media was exchanged between two clients, without being routed through the IMS network.

No. -	Time	Source	Destination	Protocol	Info
1557	253.523202	157.159.229.253	157.159.103.57	SIP	Request: MESSAGE
1558	253.551288	157.159.229.253	157.159.103.57	SIP	Status: 200 OK

Session Initiation Protocol	
+ Request-Line: MESSAGE sip:vincent@157.159.103.57:5070 SIP/2.0	
- Message Header	
Call-ID: 376e799006b757d2810d4898032809ee@157.159.103.57	
CSeq: 2 MESSAGE	
Max-Forwards: 70	
To: "vincent" <sip:vincent@siplab.int-evry.fr>;tag=554291869	
From: <sip:moshe@157.159.229.253:5070>;tag=1604172010	
Via: SIP/2.0/UDP 157.159.229.253:5070;branch=z9hg4bk4e704ce77d4480f4a7924ac3864e03de	
Contact: <sip:vincent@157.159.103.57:5070>	
Content-Type: text/plain	
Content-Length: 26	
- Message body	
Line-based text data: text/plain	
Hi kona tatana, n'wina ke? ← Note: 5	

Figure 5.13: MESSAGE Request Trace

The above results, shown in Figure 5.4 through to Figure 5.13, indicate that the developed IMS Client is capable of performing the IMS functionalities, which makes it compliant with 3GPP and IETF IMS specifications.

Though this Light-Weight IMS Client was configured to prompt the home network S-CSCF IP address before registering, registering can be performed automatically without prompting the user. This can be achieved by pre-configuring the P-CSCF or S-CSCF IP address within the Light-Weight IMS Client.

A session was established using an INVITE request. Establishing a session also initiated the SDP offer/answer mechanism to allow the two Light-Weight IMS Clients to negotiate and agree on the media type and resource reservation for QoS. It can be seen that the media attributes at the beginning of session establishment (shown in Figure 5.5) were updated regularly until the session had been established successfully.

## 5.6 LIGHT-WEIGHT AND OPEN IMS CLIENT IMPLEMENTATIONS

The implementation functionalities of the Light-Weight IMS Client and other available open IMS Clients are provided in Table 5.5.

Table 5.5: Comparison of Light-Weight and Open IMS Client Functionalities

<b>Functionality</b>	<b>Light-Weight IMS Client</b>	<b>FOKUS Open IMS Client Lite</b>	<b>IMS-Communicator</b>
Development Platform	J2ME (CLDC)	Java (JSE) and .NET	Java (JSE)
Targeted End-User Device	CLDC Compliant Mobile Phones	Pocket PC, Laptops and Desktops	Desktops and Laptops
Memory Footprint	About 37 KB (Excluding Libraries)	About 1.3 MB (Excluding Libraries)	About 1.5 MB (Excluding Libraries)
Authentication Support	In Future	Digest – AKA	Digest – AKA (Work in progress)
Client Availability	Not Available online	Available online (No source code)	Available online (With source code)
Standard Compliance	3GPP and IETF	3GPP, IETF and TISPAN	3GPP, IETF and TISPAN
SIP Support	Yes	Yes	Yes
RTP Support	In the Future	Yes	In the Future

## 5.7 CONCLUSION

In this chapter, the experimental setup for the testing of two identical Light-Weight IMS Clients within the IMS testbed was presented. The equipment and requirements for conducting the demonstration were also provided. The discussion of results was based on the SIP signalling messages exchanged between the Light-Weight IMS Clients and the IMS network (which was the IMS testbed) for the registration and establishment of the session. The SDP content was also carried within these signalling SIP messages, and used to negotiate the type of media used as well as the preconditions for the QoS agreement. The results presented shows that the Light-Weight IMS Client is compliant with the 3GPP technical specifications and IETF SIP recommendations. The IMS Client memory size is about 37 Kilo Bytes (KB) as compared to 6 305 KB of Fokus Open IMS Client (FOKUS, 2007) and 86 953 KB of UCT IMS Client version 1.0.4 (Waiting & Good, 2007). As opposed to these available IMS Clients,

the Light-Weight IMS Client presented in this project is light-weight, and can be deployed within the CLDC mobile devices which are J2ME compliant.

## CHAPTER 6

### CONCLUSION AND FUTURE WORK

#### 6.1 SUMMARY OF FINDINGS

The aim of this study was to develop a Light-Weight IMS Client that could be used to improve, qualify and finalize the specification and implementation of service discovery, and management platform for Personal Networks (PN) and the federations of PNs. Following the specifications and recommendations by the major IMS standardization bodies, these are the salient points of the findings of this study.

- To fast-track the roll-out of IMS by mobile network operators, it is important to have mobile user equipments that are compliant with the 3GPP IMS.
- J2ME is a well supported and powerful platform for the development of multimedia applications, which are suitable for small devices such as memory and power-limited mobile devices
- This study has shown that a 3GPP and IETF-compliant IMS Client can be developed using the J2ME platform for the mobile devices, hence is called the *Light-Weight IMS Client*.
- Quality of Service (QoS) between two IMS Clients was ensured by the use of SDP offer/answer mechanism during the session setup
- The IMS Client developed in this study was able to communicate with the home IMS network by registering to the IMS Testbed, establishing the session and communicating with another similar IMS Client

#### 6.2 DISCUSSION OF PROBLEMS

Although the IMS Client was developed from and based on 3GPP and IETF specifications and recommendations, it is important to note that IMS is not yet a mature technology. As a result,

most available open IMS testbed are still work in progress, hence they are unstable and not fully functional as per the 3GPP, IETF and ETSI specifications.

The IMS Client developed in this study focused on and was targeted to be tested within the INT laboratory, where an IMS testbed had been developed. Although the developed IMS Client was functional within the INT IMS testbed, it will require some modifications to perform some functions within other available IMS testbeds such as the FOKUS Open IMS testbed.

### **6.3 FINDINGS AND CONCLUSIONS**

The success of FMC in the NGN will depend on how quickly the IMS gains acceptance into the telecommunication industry. This in turn will rely on the availability of internationally standardized IMS compliant devices. This project has shown that J2ME platform can be used to develop IMS Client that is light-weight when compared to those currently available IMS Clients which are too heavy for CLDC-based mobile devices.

The demonstration results showed that the IMS Client was capable of performing the registration with the IMS network, by sending a SIP REGISTER request. The results also indicated that the IMS Client could establish a SIP session using a SIP INVITE request, negotiate QoS and media codec, using the SDP answer/offer mechanism. Finally text based message communication was established between two IMS Clients within the same IMS network. The functionalities provided by this Light-Weight IMS Client makes the client to be compliant to the 3GPP and the IETF recommendations and specifications.

### **6.4 SUMMARY OF CONTRIBUTIONS**

The contributions of this project are:

- a light-weight IMS Client that can be deployed within the Java-compliant mobile devices. This can be used for the testing of IMS services within the memory and processing power-limited mobile devices,



- the improvement and near finalization of a service discovery and management platform for PNs in the MAGNET Service Management Platform, which is the 3G and beyond platform for the Personal Networks.

## **6.5 SUGGESTIONS FOR FUTURE WORK**

For future work, the following suggestions would add more functionalities and to implementation of the IMS Client within the PN service discovery platform:

- Support for other forms of IMS services such as video sessions, audio sessions, presence service and buddy list services. At present, the IMS Client only supports text messaging media communication.
- Addition of Real-time Transport Protocol (RTP) for multimedia streaming.
- Interoperability with other 3GPP compliant open IMS networks or testbeds.
- Implementation of the IMS Client into the MAGNET service management platform for secure and context-aware IMS services discovery, session control and mobility management

## REFERENCES

- [1] 3GPP TS 23.228. June 2006. *IP Multimedia Subsystem - Stage 2*. Release 5 – version 5.15.0. Sophia Antipolis.
- [2] 3GPP TS 23.228. March 2007. *IP Multimedia Subsystem - Stage 2*. Release 6 - version 6.16.0. Sophia Antipolis.
- [3] 3GPP TS 24.008. June 2004. *Mobile radio interface layer specification: Core Network protocols –Stage 3*. Release 1999 – version 3.19.0. Sophia Antipolis.
- [4] 3GPP TR 24.930. June 2007. *Signalling flows for the session setup in the IMS based on SIP and SDP - Stage 3*. Release 7 – version 7.2.0. Sophia Antipolis.
- [5] 3GPP TS 24.228. 2006. *Signalling Flows for the IP Multimedia Call Control based on SIP and SDP - Stage 3*, Release 5. Sophia Antipolis.
- [6] 3G Americas. 2004. White paper on *IMS overview and applications*. Available from: <http://www.3g.co.uk/PR/August2004/8141.htm>. [Accessed: 04/01/2007].
- [7] 3G Americas, 2007. White Paper on *Fixed-Mobile Convergence: Understanding the marriage of wireless and wireline technologies* [Online]. [http://www.3gamericas.com/PDFs/2007\\_3GA\\_Convergence\\_White\\_Paper\\_July23.pdf](http://www.3gamericas.com/PDFs/2007_3GA_Convergence_White_Paper_July23.pdf) [Accessed: 07/08/2007]
- [8] BEA Whitepaper. 2006. *BEA Weblogic communications platform and IP Multimedia Subsystem (IMS): Next-generation converged services* [Online]. Available from: [http://www.bea.com/content/news\\_events/white\\_papers/BEA\\_IMS\\_wp.pdf](http://www.bea.com/content/news_events/white_papers/BEA_IMS_wp.pdf). [Accessed: 04/07/2007].
- [9] Bublely, D. June 2006. Report on *The evolution of SIP and IMS-capable mobile handsets* [Online]. Available from: [http://www.disruptive-analysis.com/sip\\_and\\_ims\\_handsets.htm](http://www.disruptive-analysis.com/sip_and_ims_handsets.htm) [Accessed: 23/08/2007].
- [10] Calin, D., McGee, A.R., Chandrashekhar, U. & Prasad, R. 2006. *MAGNET: An approach for secure personal networking in beyond 3G wireless networks*. Wiley Periodicals. Available from: <http://www.ist-magnet.org/>.
- [11] Camarillo, G. 2002. *SIP demystified*. McGraw-Hill.

- [12] Camarillo, G. & García-Martín, M.A. 2006. *The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the cellular worlds*. 2<sup>nd</sup> ed. John Wiley & Sons. Chichester.
- [14] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C. & Gurle, D. 2002. “*Session Initiation Protocol extension for Instant Messaging*”. IEFT, RFC 3428.
- [15] Castro, J. 2004. *All IP in 3G CDMA networks*. John Wiley and Sons. Chichester, England.
- [16] Charas, P., Cochrane, P., da Silva, J., Davenport, G, Evans, B., Eberspacher, J., Demeester, P., Haardt, M., Hara, V., Konhauser, W., Leisenburg, M., Mohr, W., Niebert, N., Nakamura, H., Pulli, P., Sydon, U., Tafazolli, R., Uusitalo, M., Verhulst, D., Walke, B., Welens, U & Zorzi, M. (eds.). 2000. “*The book of visions 2000 – visions of the wireless world*”. Version1.0. Wireless Strategic Initiative, IST-WSI Project. [Online] Available from: [http://www.wireless-world-research.org/general\\_information/Bookofvisions/BookofVisions2000.html](http://www.wireless-world-research.org/general_information/Bookofvisions/BookofVisions2000.html). [Accessed: 09/12/2007].
- [17] Chen, R.L.J., Su, E.C.Y., Shen, V.S.C. & Wang, Y. 2006. IBM Whitepaper on *Introduction to IP Multimedia Subsystem part 1: SOA Parlay X Web services*. Available from: <http://www-128.ibm.com/developerworks/webservices/library/ws-soa-ipmultisub1/>. [Accessed: 04/01/2007].
- [18] Ericsson. 2004. White paper on IMS: *The value of using IMS architecture* [Online]. Available from: [http://www.ericsson.com/technology/whitepapers/ims\\_ip\\_multimedia\\_subsystem.pdf](http://www.ericsson.com/technology/whitepapers/ims_ip_multimedia_subsystem.pdf). [Accessed: 04/01/2007]
- [19] ETSI TR 180.001 V1.1.1. 2006. *Protocols for Advanced Networking (TISPAN)*, NGN Release 1. Available from: <http://www.etsi.org>.
- [20] Fabini, J., Reichl, P., Poropatich, A., Huber, R. & Jordan, N. 2006. *IMS in a Bottle: Initial Experiences from an OpenSER-Based prototype implementation of the 3GPP IP Multimedia Subsystem*. In: Proceedings of the International Conference on Mobile Business (ICMB'06), IEEE.
- [21] FOKUS. *Open IMS Client*. Available from: [http://www.fokus.fraunhofer.de/ims/components/open\\_ic.php?lang=en](http://www.fokus.fraunhofer.de/ims/components/open_ic.php?lang=en). [Accessed: 25/5/2007]

- [22] Garcia-Martin, M., Henrikson, E. & Mills, D. 2003. *Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)*. IETF, RFC 3455.
- [23] Goyal, V. 2005. *J2ME Tutorial, Part 1: Creating MIDlets* [Online]. Available from: <http://today.java.net/pub/a/today/2005/02/09/j2me1.html?> [Accessed: 19/02/2007].
- [24] Handley, M., Jacobson, V. & Perkins, C. July 2006. *SDP: Session Description Protocol*. IETF, RFC 4566.
- [25] Hills, T. 2006. Technical Report: “*Tispan: IMS Plus*” [Online] Available from: [http://www.lightreading.com/document.asp?doc\\_id=95511&page\\_number=2](http://www.lightreading.com/document.asp?doc_id=95511&page_number=2). [Accessed: 11/05/07].
- [26] JCP JSR 180. 2004. *SIP API for Java 2 Micro Edition Specification*. Version 1.0.1. Available from: <http://www.jcp.org>.
- [27] JCP JSR 248. September 2006. *Mobile architecture services specification*. Version 1.00. Available from: <http://www.jcp.org>.
- [28] Jennings, C., Peterson, J. & Watson, M. 2002. *Private Extensions to the Session Initiation Protocol for Asserted Identity within Trusted Networks*. IETF RFC 3325.
- [29] Johnston, A.B. 2001. *SIP: Understanding the session initiation protocol*. Artech House, London.
- [30] Li, S. & Knudsen, J. 2005. *Beginning J2ME: from novice to professional*. 3<sup>rd</sup> Ed. Apress.
- [31] Lin, Y., Chang, M., Hsu, M. & Wu, L. 2005. *One-pass GPRS and IMS authentication procedure for UMTS*. The journal on selected areas in communications, vol. 23, No. 6. The IEEE, June: 1233 – 1239.
- [32] Magedanz, T., Witaszek, D. & Knuettel, K. 2005. *The IMS playground @FOKUS – An open testbed for next generation network multimedia services*. In: Proceedings of the First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities. IEEE Computer Society.
- [33] MAGNET. 2006. *Introduction to MAGNET* [Online]. Available from: <http://www.ist-magnet.org/technicalapproach>. [Accessed: 29 November 2006].
- [34] Mao, G.F., Talevski, A. & Chang, E. 2007. *Voice over Internet Protocol on mobile devices*, IEEE Computer Society, Proceedings on the 6<sup>th</sup> IEEE/ACIS International Conference on Computer and Information Science.

- [35] Marshall, W. 2003. *Private Session Initiation Protocol (SIP) Extensions for Media Authorization*. IETF, RFC 3313.
- [36] Newport Networks. 2005. Whitepaper on *Session border control functions in IMS-based converged networks* [Online]. Available from: <http://newport-networks.com>. [Accessed: 12/02/2007]
- [37] O'Doherty, P. June 2003. *SIP and the Java platforms*, Sun White Paper [Online], Available from: <http://java.sun.com/products/jain/SIP-and-Java.html>. [Accessed: 19/02/2007].
- [38] Poikselkä, M., Mayer, G., Khartabil, H. & Niemi, A., 2004. *The IMS: IP Multimedia Concepts and services in the mobile domain*. John Wiley & Sons, West Sussex.
- [39] Poikselkä, M., Mayer, G., Khartabil, H. & Niemi, A. 2006. *The IMS: IP multimedia concepts and Services*, 2<sup>nd</sup> ed. John Wiley & Sons, West Sussex.
- [40] Prasad, R. & Skouby, K. 2005. *Personal network applications* [Online]. Available from: <http://www.ist-magnet.org/GetAsset> [Accessed: 09/12/2006].
- [41] PT Inovação. 2007. *IMS-Communicator* [Online], Available from: <http://developer.berlios.de/projects/imscommunicator/> [Accessed: 01/11/2007]
- [42] RADVISION. 2006. Whitepaper on *IMS SIP and Signaling: The RADVISION perspective* [Online]. Available from: <http://www.radvision.com>. [Accessed: 05/01/2007].
- [43] Rosenberg, J. 2002. *The SIP UPDATE method*. RFC 3311. IETF.
- [44] Rosenberg, J. & Schulzrinne, H. June 2002. *An Offer/Answer model with the Session Description Protocol (SDP)*. RFC 3264. IETF
- [45] Rosenberg, J. & Schulzrinne, H. 2002. *Reliability of provisional response in the Session Initiation Protocol (SIP)*, RFC 3262, IETF
- [46] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. & Schooler, E., 2002, *SIP: Session Initiation Protocol*, IETF, RFC 3261.
- [47] Safavian, S.R. 2006. *IMS: A standardized approach to all-IP converged network*. [Online]. Available from: [http://bechtelecoms.com/docs/bttj\\_v4nl/Article2.pdf](http://bechtelecoms.com/docs/bttj_v4nl/Article2.pdf) [Accessed: 25/01/2007].
- [48] Schulzrinne, H. & Volz, B. July 2003. *Dynamic Host Configuration Protocol (DHCPv6) options for SIP servers*. RFC 3319. IETF
- [49] Soliman, H. 2004. *Mobile IPv6: Mobility in a wireless internet*. Addison-Wesley, Paris.

- [50] Waiting, D. & Good, R. 2007. *UCT IMS Client* [Online]. Available From: <http://uctimsclient.berlios.de/>. [Accessed: 01/11/2007]
- [51] Wong, K.D. & Varma, V.K. 2003. *Supporting Real-Time IP Multimedia Services in UMTS*. IEEE Communications Magazine. pp. 148 – 155.

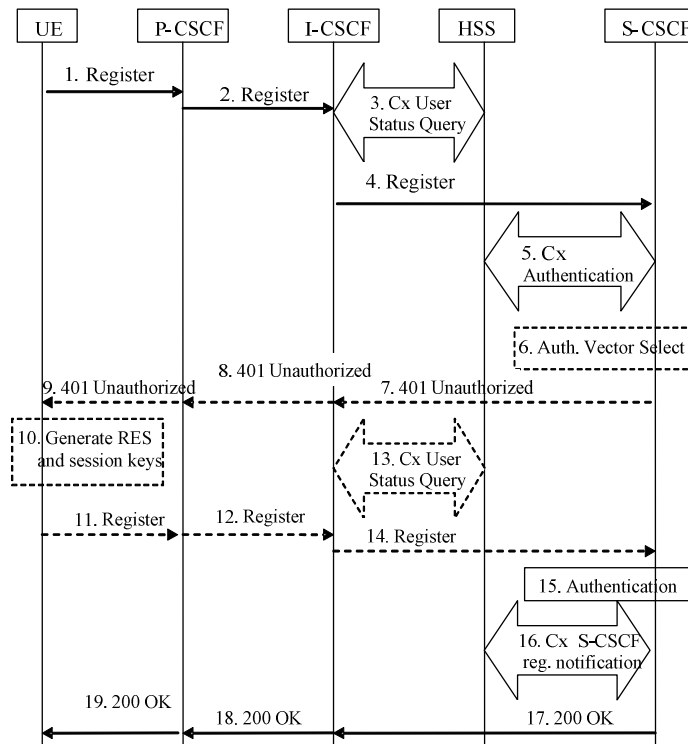
## APPENDIX A

### SIP MESSAGE FLOW

The SIP message flow shown in Appendix A was adapted from the 3GPP TS24.228 Release 6.

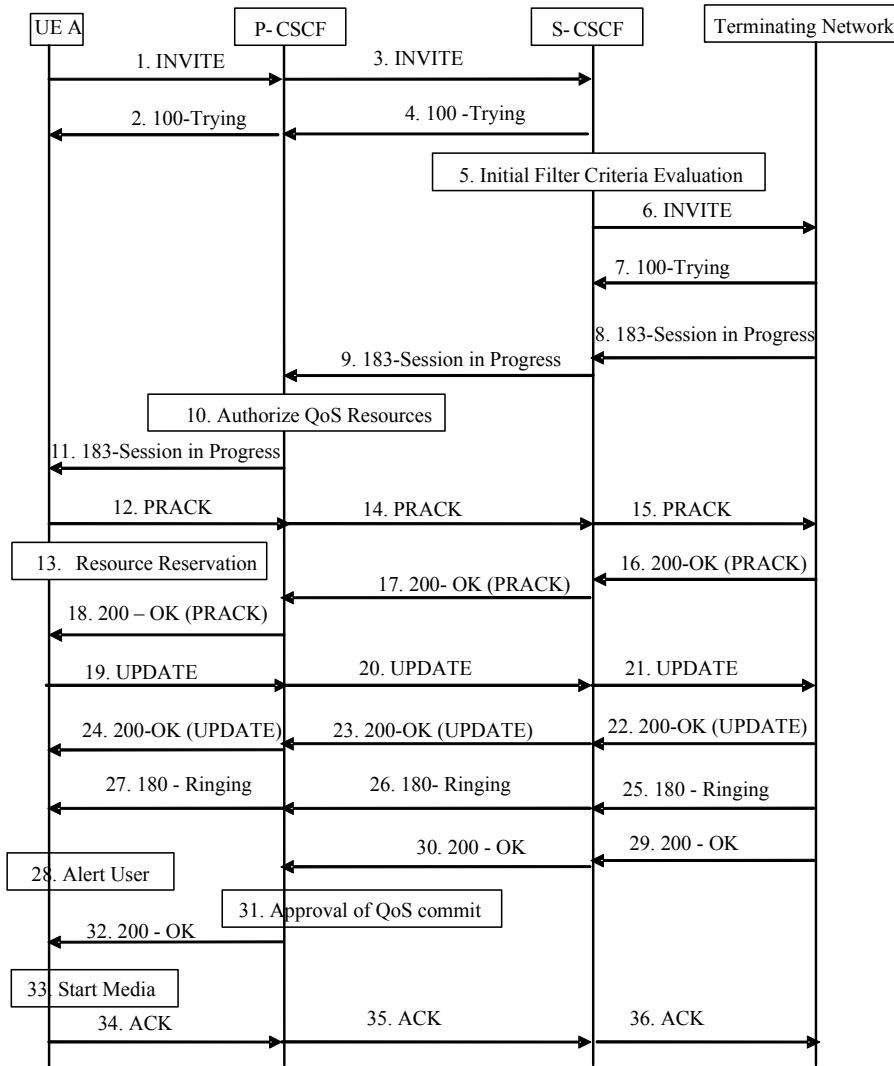
#### A.1: SIP Registration Message Flow

The dotted lines in show messages and procedures to be followed based on the 3GPP specification. These dotted line functionalities are not supported by the INT IMS testbed yet, therefore they will not be implemented by this IMS Client. As the INT IMS testbed is still work-in-progress, such functionalities will be supported in future. The IMS Client will be registered only when the initial SIP REGISTER request has been sent, as shown by the solid lines.



#### A.2: Session Establishment Message Flow

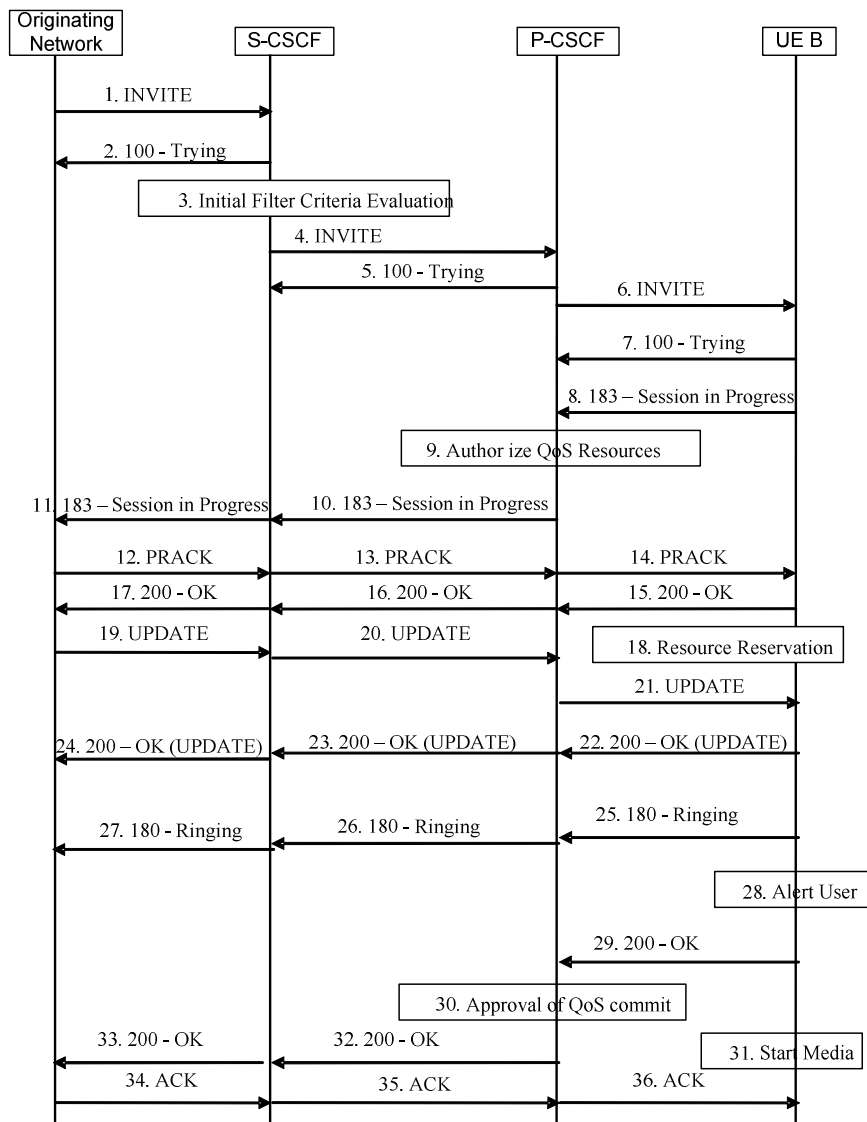
Appendix A.2 is the sequence diagram indicating the establishment of the session from UE A to the terminating network (3GPP TR 24.930, 2007). The terminating network, which belongs to UE B, is shown in Appendix A.3. An I-CSCF is not shown in the diagram. It is important to note that the terminating network can also be the same network, if both UEs are served by the same IMS network operator.





### A.3: Session Termination Message Flow

Appendix A.3 is the sequence diagram indicating the termination of the session from the originating network to UE B (3GPP TR 24.930, 2007). The originating network may be similar to the one shown in Appendix A.2, but it can also share the same network with the terminating network.



## APPENDIX B

### JAVA CODE TO CREATE A DISPLAY OF J2ME

```
//Create the commands to be attached to the register display
protected Command exitCmd = new Command("Exit", Command.EXIT,1);
protected Command registerCmd= new Command("Register", Command.OK,0);

//Create registerFrm: Allow the user to register
private Form getRegisterFrm() {
    if (registerFrm == null){
        registerFrm = new Form("Unregistered to IMS",
            new Item[] { new TextField("You have to register
first\nPress Register to register\n\n Registrar IP
address:" ,registrar, 30,      TextField.ANY) });

        registerFrm.addCommand(exitCmd);
        registerFrm.addCommand(registerCmd);
        registerFrm.setCommandListener(this);
    }
    return registerFrm;
}
```

## APPENDIX C

### JAVA CODE

The IMS Client was written in a Java code, and made an extensive use of the JSR 180 SIP API and other new Java classes created to work on top of this API. The following Java classes are included on the attached CD:

- HostSDP.java: stores the SDP message of the host IMS Client.
- PrackRequest.java: generates and processes the SIP PRACK request.
- SDP\_Msg\_Parser.java: provides parsing for SDP content.
- StringSplit.java: provides the splitting of the strings in SDP.
- StringUp.java: recombines the strings.
- TheTime.java: provides the timestamp during simulation.
- UpdateRequest.java: generates and processes the SIP UPDATE request.
- Vincent\_IMS.java: is the main IMS Client class.