

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]





uOttawa

L'Université canadienne
Canada's university

**FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTDOCTORALES**



**FACULTY OF GRADUATE AND
POSTDOCTORAL STUDIES**

Majdi Rawashdeh

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.

GRADE / DEGREE

School of Information Technology and Engineering

FACULTE, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Seamless Video Handoff in Session Mobility over the IMS Network

TITRE DE LA THÈSE / TITLE OF THESIS

A. Karmouch

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

M. Barbeau

T. Yeap

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Seamless Video Handoff in Session Mobility over the IMS Network

by

Majdi Rawashdeh

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

Master of Computer Science

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-59461-2
Our file *Notre référence*
ISBN: 978-0-494-59461-2

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Abstract

Enabling service continuity from one user's device to another while maintaining communications a justifiable ambition; the benefits reaped are maximized end-user service experience and the opportunity to employ the capacities of all devices in the user's environment. Such ubiquity of service can be realized by session mobility between devices.

Providing seamless handoff in session mobility is a challenging, and vital, issue relating to the heterogeneous next generation All-IP networks. The research work outlined by this thesis describes a solution to support seamless video handoff in session mobility over IP Multimedia Subsystem (IMS) networks.

Our system is built on top of an IMS network, which we believe to be the best platform for session mobility deployment, as it merges mobility and Internet services. The system combines two techniques to achieve seamless video handoff: It uses the SIP REFER method, with a proposed Prediction Handoff Manager that predicts the required handoff time to transfer the session completely.

The proposed system is implemented using PJSIP framework, an open source of SIP. Results show effective seamless handover in session mobility over IMS networks.

Acknowledgements

Firstly, I must express my deepest gratitude to my supervisor, Professor Ahmed Karmouch; his valuable guidance, constructive advice, academic support and encouragement have been vital to the progress of my research and completion of this thesis.

Special thanks to my colleague Damien Herraud for his invaluable help, support and effort during the implementation phase.

Thanks are also due to Stenio Fernandes, Ibrahim Al-Oqily and other members of the Intelligence for Mobile Autonomic and Cognitive Networks Laboratory, for their suggestions and cooperation during my research work.

Last, but definitely not least, I should express my boundless gratitude and love to my family: My wife Asma and my parents whose consistent encouragements, support and endless love have taken me through all hardships incurred during my studies and research journey.

Table of Contents

ABSTRACT	II
ACKNOWLEDGEMENTS	III
TABLE OF CONTENTS	IV
LIST OF FIGURES	VI
LIST OF TABLES	VII
LIST OF ABBREVIATIONS	VIII
CHAPTER 1 INTRODUCTION	1
1.1 BACKGROUND AND MOTIVATION	1
1.2 OBJECTIVES AND CONTRIBUTION	3
1.3 THESIS ORGANIZATION	4
CHAPTER 2 BACKGROUND	6
2.1 OVERVIEW OF SIP.....	6
2.1.1 <i>Entities and SIP addresses</i>	7
2.1.2 <i>Methods</i>	9
2.1.3 <i>Message</i>	10
2.1.4 <i>SIP session establishment</i>	13
2.1.5 <i>Sip extensions</i>	15
2.1.6 <i>Applications and Services</i>	16
2.2 IP MULTIMEDIA SUBSYSTEM	20
2.2.1 <i>IMS Overview</i>	20
2.2.2 <i>IMS Architecture</i>	21
2.2.3 <i>IMS benefits</i>	28
2.2.4 <i>IMS Applications</i>	32
2.2.5 <i>IMS Message Flow</i>	35
2.3 SUMMARY	37
CHAPTER 3 MOBILITY MANAGEMENT	38
3.1 OVERVIEW	38
3.2 TYPES OF MOBILITY	40
3.2.1 <i>Personal Mobility</i>	40
3.2.2 <i>Terminal Mobility</i>	42
3.2.3 <i>Service Mobility</i>	42
3.3 SESSION MOBILITY	43
3.3.1 <i>Techniques</i>	43
3.3.2 <i>Transfer modes</i>	44
3.3.3 <i>Types of transferred media</i>	47
3.3.4 <i>Seamless handoff requirements</i>	48
3.4 RELATED WORK	49
3.5 SUMMARY	52
CHAPTER 4 ARCHITECTURE AND FEATURES	53

4.1 INTRODUCTION.....	53
4.2 ARCHITECTURE OVERVIEW.....	54
4.3 SYSTEM CLIENTS DESCRIPTION	55
4.3.1 <i>Server Handoff Prediction Manager</i>	55
4.3.2 <i>Client Buffer Controller</i>	57
4.4 HANDOFF SCENARIO	57
4.5 SYSTEM ELEMENTS DESCRIPTION.....	58
4.5.1 <i>Call Flow of Session Transfer</i>	58
4.5.2 <i>Mobile Node (MN)</i>	62
4.5.3 <i>Recipient Node (RN)</i>	64
4.5.4 <i>Correspondent Node (CN)</i>	66
4.6 SUMMARY	68
CHAPTER 5 SYSTEM IMPLEMENTATION	69
5.1 OVERVIEW	69
5.2 TOOLS	70
5.2.1 <i>PJSIP Framework</i>	70
5.2.2 <i>Open IMS Core</i>	70
5.2.3 <i>Gstreamer Framework</i>	70
5.2.4 <i>VLC Streaming Server</i>	71
5.3 SYSTEM DEPLOYMENT	72
5.3.1 <i>Open IMS Core deployment</i>	72
5.3.2 <i>PJSIP deployment</i>	73
5.4 EXPERIMENTAL ENVIRONMENT	75
5.5 WORKING PROTOTYPE DESCRIPTION	76
5.5.1 <i>Registering to IMS</i>	77
5.5.2 <i>Adding buddy list</i>	79
5.5.3 <i>Establishing video session</i>	80
5.5.4 <i>Transferring video session</i>	80
5.6 PERFORMANCE EVALUATION	82
5.7 SUMMARY	84
CHAPTER 6 CONCLUSION AND FUTURE WORK.....	85
6.1 CONCLUSION.....	85
6.2 FUTURE WORK AND SUGGESTIONS	87
BIBLIOGRAPHY	89

List of Figures

Figure 2.1 SIP Message Format.....	10
Figure 2.2 SIP Message Example	13
Figure 2.3 SIP Session Establishment	15
Figure 2.4 Home telephone service using SIP	17
Figure 2.5 IM programs using SIP	18
Figure 2.6 Controlling Devices using SIP	20
Figure 2.7 IMS Architecture	22
Figure 2.8 Example of Registration to IMS.....	36
Figure 3.1 SIP support Mobility	40
Figure 3.2 Personal Mobility	41
Figure 3.3 Third party call control for session mobility	45
Figure 3.4 SIP REFER Method for Session Mobility.....	47
Figure 4.1 Architecture workflow.....	55
Figure 4.2 Video Transfer Scenario.....	58
Figure 4.3 Call Flow of Session Transfer	61
Figure 4.4 MN behaviour.....	63
Figure 4.5 RN behaviour.....	65
Figure 4.6 CN behaviour.....	67
Figure 5.1 VLC Main Window.....	71
Figure 5.2 VideoIMS main screen	73
Figure 5.3 Experimental Environment.....	75
Figure 5.4 Registering to IMS network	78
Figure 5.5 Adding CN, RN to buddy list.....	79
Figure 5.6 Establishing video session.....	80
Figure 5.7 Transferring the video session.....	81
Figure 5.8 Predicted Handoff Time vs. Actual	83
Figure 5.9 Differences between Predicted and Actual Handoff Time.....	84

List of Tables

Table 2.1 SIP Methods.....	9
Table 2.2 Status Codes.....	10

List of Abbreviations

3G	Third Generations
3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
API	Application Programming Interface
ATA	Analogue Telephone Adaptor
B2BUA	Back-to-Back User Agent
CN	Correspondent Node
CS	Circuit-Switched
CSCF	Call Session Control Function
DMP	Device Messaging Protocol
GPRS	General Packet Radio Service
GUI	Graphical User Interface
HPM	Handoff Prediction Manager
HSS	Home Subscriber Server
HTTP	Hyper Text Transfer Protocol
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
MIME	Multipurpose Internet Mail Extensions
MN	Mobile Node
MPEG	Moving Picture Experts Group
MRF	Media Resource Function
NAT	Network Address Translation
NGN	Next Generation Networking
PBX	Private Branch Exchange
PDA	Personal Digital Assistant

PDF	Policy Decision Function
PoC	Push-To-Talk over Cellular
PS	Packet-Switched
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFC	Request for Comment
RN	Recipient Node
RTP	Real time Protocol
SDP	Session description protocol
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SMS	Short Message Service
SSP	Subscriber User Profile
TCP	Transmission Control Protocol
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol

Chapter 1 Introduction

1.1 Background and Motivation

The ever-increasing computing power of mobile devices has allowed users, wherever they are, to enjoy many enhanced features of media on the Internet. However, these mobile devices still remain less powerful than a classical desktop computer. The mobile device could be merged with the comfort of commodity hardware by facilitating a user's ability to switch transparently from one device to another without interrupting his/her current activity.

Mobility management is an important issue for a mobile device, and session initiation protocol (SIP) is an excellent candidate as a protocol for mobility management in All-IP networks [13], as it has the ability to support not only terminal mobility but also personal mobility, session mobility, and service mobility. Session Mobility refers to the user's ability to maintain an active session while switching between terminals.

Since mobile phones are increasingly powerful and now able to connect to the Internet almost everywhere in the world, IP Multimedia Subsystem (IMS) seems to be the best platform for session mobility deployment, as it merges mobility and internet services

CHAPTER 1. INTRODUCTION

such as video streaming. IMS is defined by 3rd Generation Partnership Project (3GPP) as a new subsystem, i.e., a new mobile network infrastructure that enables the convergence of data, speech, and mobile network technology over an IP-based infrastructure. IMS uses the SIP protocol for multimedia session negotiation and session management. It allows devices from different networks to be connected. This interoperability is very pertinent in session mobility, where a video will be streamed and transferred to devices that may belong to different heterogeneous networks.

Therefore, providing seamless handoff in session mobility is one of the most challenging problems in the heterogeneous next generation All-IP networks; the existing solutions for session mobility do not provide an optimal solution to handle the disruption time, which indicates the tear-down of the existing media stream and the establishment of the new stream. During this period the user feels a lapse or gap in the communication, in addition to the loss of media frames.

Our proposed framework is designed to minimize and decrease the disruption time by transferring the video session seamlessly without data or frame loss and with minimum amount of delay; in such away the whole video transfer process is transparent to the user.

CHAPTER 1. INTRODUCTION

1.2 Objectives and Contribution

The objective of the thesis is to design a system that is capable of transferring a video session seamlessly from one device to another one over the IMS networks. The system proposes a SIP-based Handoff Prediction Manager to determine the size of the video to be streamed to both, the Mobile Node and the Recipient Node, during the handoff time. It accomplishes that task by predicting the required handoff time needed to transfer the video session completely.

The objectives of the thesis are summarized as follows:

- Examine a list of requirements for the proposed system and fulfill them.
- Propose an approach to effectively transfer the video session seamlessly.
- Apply the proposed approach to a prototype and validate it.

The main contributions of the thesis are as follows:

(i) Conception and implementation of a SIP-based Handoff Prediction Manager.

The thesis proposes a design that uses SIP and other technologies to transfer the video session seamlessly in session mobility. The proposed architecture assumes that all participating devices are aware of each other.

(ii) Definition and realization of the features of SIP user agent (VideoIMS) over the

IMS network to enhance the system usability. An effective approach to transfer the video session is described in this thesis. Session participants can launch VideoIMS to start the session, add other participants to the buddy list and transfer the session seamlessly.

CHAPTER 1. INTRODUCTION

(iii) Configuration and deployment of IP Multimedia Subsystem network.

1.3 Thesis Organization

The remainder of thesis report is organized as follows:

Chapter 2 lays the ground to the thesis by introducing background knowledge of SIP and IMS. Firstly, an overview of SIP is provided, where its architecture and major components, session establishment scenario and extensions are described in detail. Next, an introduction to the IMS system: The architecture, applications, benefits and message flow within the system are presented.

Chapter 3 explores some issues regarding mobility management. Starting by an overview of mobility types including service, terminal and personal mobility, the chapter then proceeds to discuss session mobility. In the latter, we introduce the main techniques, transfer modes, types of transferred media and requirements for seamless handoff in session mobility. Finally, we review some of the related work in session mobility.

In Chapter 4 we present the system architecture overview, and a detailed description of the elements and clients involved in this system. We then proceed with a Handoff scenario to explain our proposed system; we also present the call flow of session transfer and the behavior of the each element in the system.

Chapter 5 presents a prototype implementation of our proposed SIP-based Handoff Prediction system. The tools and techniques for implementation are introduced in details.

CHAPTER 1. INTRODUCTION

The chapter also presents selected snapshots and figures illustrating the discussed prototype.

Finally, Chapter 6 concludes this thesis and discusses our future research plans and potential enhancements to our current design.

Chapter 2 Background

This chapter provides background information related to the thesis. As Session Initiation Protocol (SIP) and IP Multimedia Subsystem (IMS) are involved in the design of our proposed framework, they are discussed in detail in this chapter.

Firstly, an overview of SIP is provided. The architecture, its major components, session establishment scenario and extensions of SIP are introduced in detail. This is followed by an introduction to IMS system; its architecture, applications, benefits and message flow within it.

2.1 Overview of Sip

The Session Initiation Protocol (SIP) [25][15] is an agile, general-purpose signaling protocol for creating, modifying, and terminating multimedia sessions or calls. It has been standardized within IETF [14] as a signaling protocol for establishing real-time calls and conferences over Internet. It may be utilized in applications such as multimedia conferences, distance learning and Internet telephony. SIP is a text-based protocol and can be easily extended. Since SIP is a general purpose protocol, it works independently of underlying transport protocols and without dependency on the type of session that is

CHAPTER 2. BACKGROUND

being established. SIP incorporates with other protocols for multimedia communications and control. For example, it incorporates with Session Description Protocol (SDP) [32] for multimedia session description during SIP session establishment and it incorporates with real-time transport protocol (RTP) [16] for real-time data transportation after SIP session establishment. SIP is an application layer signaling control protocol and it does not provide services, but SIP provides primitives that can be used to implement different services. Recently, SIP has been implemented by a numerous number of vendors, especially in the area of Internet Telephony (VoIP) [47].

2.1.1 Entities and SIP addresses

The main entities in SIP are User Agents, Registrars, SIP Proxy Servers, and SIP Redirect Servers.

The SIP User Agent (UA) also known as SIP endpoint, functions as client namely UAC (User Agent Client) when initiating requests and functions as server namely UAS (User Agent Server) when responding to requests. UA communicates with other UA's directly or via SIP Proxy or via Redirect Servers. SIP Proxy and Redirect Server provide different working modes in a session establishment. When a SIP UA request is received, SIP Proxy forwards this request to the next SIP Server or another UA within the network. A SIP Redirect Server however returns the requested servers' address to UA and the UA contacts the destination directly. A SIP Registrar is responsible for maintain the location service, namely storage of SIP UA addresses. The Proxy and Redirect Servers check the destination SIP UA's address from location service to help establish session between caller and callee. The SIP Proxy, Redirect Servers, and Registrar are logical entities which

CHAPTER 2. BACKGROUND

may physically exist in one machine. More details of these entities can be found in [25][15].

Every SIP entity has a unique SIP address to identify them; SIP address is presented in the form of SIP URL:

➤ *sip: username@domain*

The user part is either a user name or telephone number. The domain part is either a domain name or a numeric network address. While the user part is optional, the domain name is mandatory. The domain name may include parameters like user name, port number, etc. Some examples of SIP addresses are:

➤ *sip: alice@uottawa.ca*

➤ *sip:+1-613-562-5800:6373@uottawa.ca: user = phone*

➤ *sip:alice@192.0.100.2*

Sip address is a globally reachable address. SIP users are bound to this address by frequently registering their latest location or address to SIP Registrar server. The location service will be looked up whenever a caller wants to establish real-time communication with callee.

Besides using SIP URL, SIP users may also use e-mail addresses or telephone numbers as their additional aliases. SIP UA's can be identified or located by their aliases as well.

CHAPTER 2. BACKGROUND

2.1.2 Methods

SIP defines an extensible set of request methods listed in Table 2.1, more detail of these methods can be found in [25]

Table 2.1 SIP Methods

Method name	Meaning
<i>ACK</i>	Acknowledges the establishment of a session
<i>BYE</i>	Terminates a session
<i>CANCEL</i>	Cancels a pending request
<i>INVITE</i>	Establishes a session
<i>NOTIFY</i>	Notifies the user agent about a particular event
<i>PRACK</i>	Acknowledges the reception of a provisional response
<i>REGISTER</i>	Maps a public URI with the current location of the user
<i>SUBSCRIBE</i>	Requests to be notified about a particular event
<i>MESSAGE</i>	Carries an instant message
<i>REFER</i>	Instructs a server to send a request

The most important method is INVITE method. SIP users initiate a call by issuing an INVITE request. When a caller wants to have a session with a callee, his/her SIP UA sends out SIP INVITE to the particular callees' SIP UA. IF the callee accepts the call, his/her SIP UA sends back a 200 OK response to the caller. After getting the 200 OK responses, caller generates an ACK request to callee to confirm the session establishment. This finishes the process of session establishment and user can now use other protocol like RTP for real-time media communication. A new INVITE request will create a new session. SIP may change an existing session by re-issuing a new INVITE. The REGISTER request is used to update a users' current contact address so by sending a REGISTER request to SIP Registrar , SIP UA may update their latest reachable location and address in SIP server.

CHAPTER 2. BACKGROUND

2.1.3 Message

SIP is based on HTTP [38] and, so, is a textual request-response protocol. In SIP, requests and responses between clients and servers are called SIP messages. Figure 2.1 shows the format of SIP messages. They start with the *start line*, which is called the *request line* in requests and the *status line* in responses. The *start line* is followed by a number of header fields that follow the format *name: value* and an empty line that separates the header fields from the optional message body.

Start line
A number of header fields
Empty line
Optional message body

Figure 2.1 SIP Message Format

The *status line* contains the protocol version and the status of the transaction, which is given in numerical (status code) as shown in Table 2.2 and in human-readable (reason phrase) formats. Here is an example for the OK response (when a request is accepted):

➤ *SIP/2.0 200 OK*

Table 2.2 Status Codes

Status code range	Meaning
100 - 199	Provisional
200 - 299	Success
300 - 399	Redirection
400 - 499	Client error
500 - 599	Server error
600 - 699	Global failure

CHAPTER 2. BACKGROUND

The *start line* in requests is referred to as the *request line*. It contains a method name, the Request-URI, and the protocol version. The method name indicates the purpose of the request and the Request-URI represents the destination of the request. Below is an example:

➤ *INVITE sip:alice@open-ims.test SIP/2.0*

Such as HTTP messages, SIP messages (both requests and responses) contain a set of header fields. There are mandatory header fields that appear in every message and optional header fields that only appear when needed. A header field consists of the header field's name, a colon, and the header field's value, as shown in the example below:

➤ *To: Alice <sip:alice@open-ims.test>;tag=1234*
From: Bob <sip:bob@open-ims.test>
Route: <sip:proxy.open-ims.test>

The six mandatory header fields are summarized below:

To: the To header field contains the URI of the destination of the request. However, this URI is not used to route the request. It is intended for human consumption and for filtering purposes.

From: the From header field contains the URI of the originator of the request. Like the To header field, it is mainly used for human consumption and for filtering purposes.

Cseq: the Cseq header field contains a sequence number and a method name. They are used to match requests and responses.

Call-ID: the Call-ID provides a unique identifier for a SIP message exchange.

CHAPTER 2. BACKGROUND

Max-Forwards: the Max-Forwards header field is used to avoid routing loops. Every proxy that handles a request decrements its value by one, and if it reaches zero, the request is discarded.

Via: the Via header field keeps track of all the proxies a request has traversed. The response uses these Via entries so that it traverses the same proxies as the request did in the opposite direction.

The message body describes individual media sessions of the call and it is separated from the header fields by an empty line. SIP messages can carry any type of body and even multipart bodies using MIME (Multipurpose Internet Mail Extensions) encoding [35]. SIP uses MIME to encode its message bodies. Consequently, SIP bodies are described in the same way as attachments to an email message. A set of header fields provide information about the body: its length, its format, and how it should be handled. For example, the header fields below describe the SDP session description:

➤ *Content-Disposition: session*
Content-Type: application/sdp
Content-Length: 193

The Content-Disposition indicates that the body is a session description; the Content-Type indicates that the session description uses the SDP format, and the Content-Length contains the length of the body in bytes.

SIP uses SDP for media description, which conveys information in textual format. In the media description, users can negotiate *media type* and *encoding* used for communication.

CHAPTER 2. BACKGROUND

SIP has no restrictions to any particular media type. A multimedia session can contain any media streams, such as audio, video, text, etc. An example SIP INVITE message and media description of SDP are given as follows:

```
INVITE sip:video@open-ims.test SIP/2.0
Via: SIP/2.0/UDP 137.122.0.1:5060;branch=z9hG4bKna43f
Max-Forwards: 70
From: < sip:pc@open-ims.test >;tag=1234
To: < sip:video@open-ims.test >
Call-ID: 6328776298220188511@137.122.0.1
Cseq: 1 INVITE
Content-Type: application/sdp
Content-Length: 193

v=0
o=pc 2890844526 2890844526 IN IP4 open-ims.test
s= hi
c=IN IP4 137.122.0.1
t=0 0
m=audio 20000 RTP/AVP 0
```

Figure 2.2 SIP Message Example

2.1.4 SIP session establishment

In this example Bob invites Alice to an audio session. Figure 2.3 shows the establishment of the audio session between Bob and Alice through the proxy server at *open-ims.test*. At the beginning Bob sends an INVITE request using Alice's public URI *sip:alice@open-ims.test* as the *Request-URI*. The proxy at *open-ims.test* relays the INVITE request to Alice at her current location (her PDA 137.122.0.1). Alice accepts the invitation sending a 200 (OK) response, which is relayed by the proxy to Bob.

As we see Alice has included a Contact header field in her 200 (OK) response. This header field is used by Bob to send subsequent messages to Alice. This way, once the

CHAPTER 2. BACKGROUND

proxy at *open-ims.test* has helped Bob locate Alice, Bob and Alice can exchange messages directly between them.

Bob uses the URI in the Contact header field of the 200 (OK) response to send his ACK.

Now that the audio session is established, Bob and Alice can talk whenever they want.

If, in the middle of the audio session, they wanted to make any changes to the session for example adding video, all what they need is to issue another INVITE request with an updated session description. INVITE requests sent within an ongoing session are usually referred to as re-INVITEs.

When Bob and Alice finish their conversation, Bob sends a BYE request directly to Alice, without the intervention of the proxy. Alice responds with a 200 (OK) response to the BYE request.

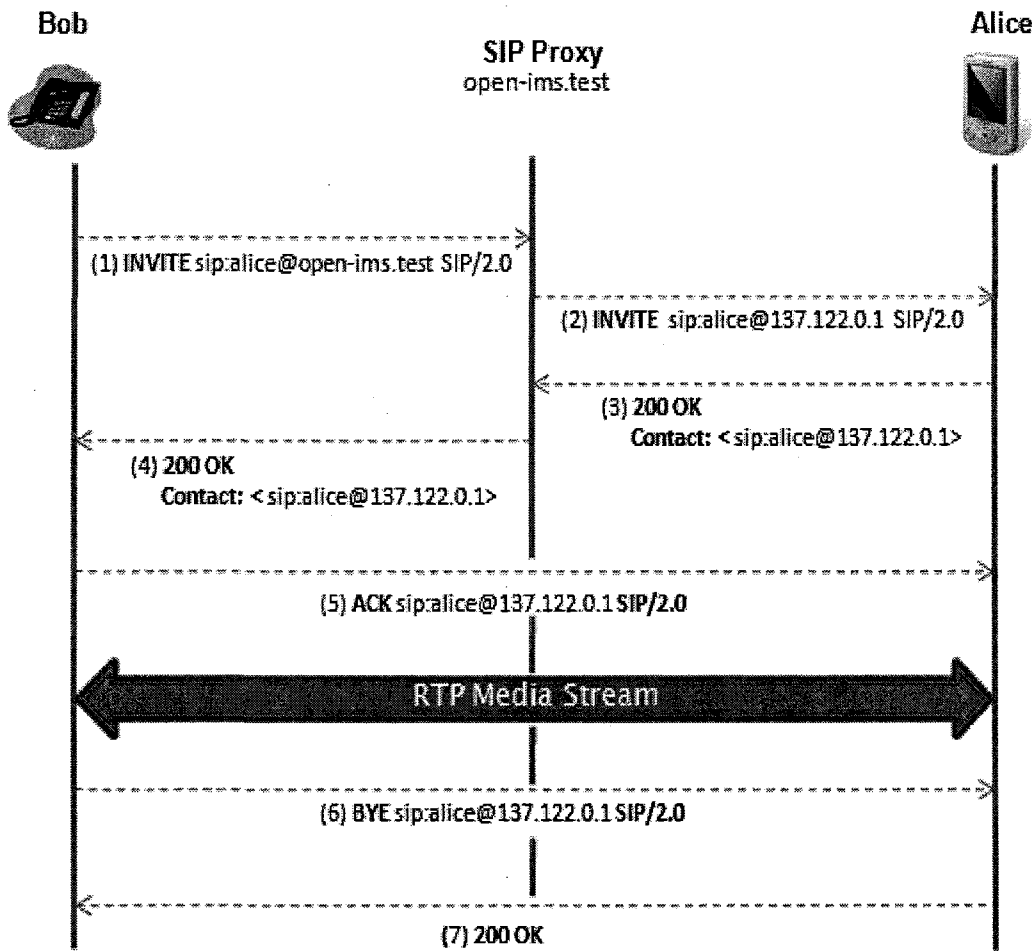


Figure 2.3 SIP Session Establishment

2.1.5 Sip extensions

SIP is a texted-based protocol, just like HTTP. It can be extended to add new methods or header fields if new functionality or modification is needed [9][24]. Integrity and confidentiality properties of such methods or header fields have to be described by the creator. There exist SIP extensions to deliver instant messages and to handle subscriptions to events. SIP has been widely used as a VoIP signaling protocol because of its extensibility.

CHAPTER 2. BACKGROUND

2.1.6 Applications and Services

This section views some of the most attractive applications and services that SIP facilitates:

1. *Internet Telephony*: SIP will bring Internet telephony to users' desktops; it's the next evolutionary step after VoIP, networks, and IP Telephony. Today, two main reasons drive organizations and to employ internet telephony - it lowers the cost of network ownership and enhances business communications via rapid deployment of new applications. The cost of network ownership consists of equipment, maintenance, network administration and network carrier costs; all of which are significantly reduced if IP internet telephony is adopted rather than the existing PBX solutions. Similarly, Internet telephony enhances communication by enabling the workforce to be more mobile hence more productive as well as enabling rich multimedia communication such as videoconferencing at an incredibly low cost.

Internet Telephony does not target businesses only; it attracts millions of home subscribers as well. Figure 2.4 below shows a system that VONAGE™, a telephone service provider can install at your home

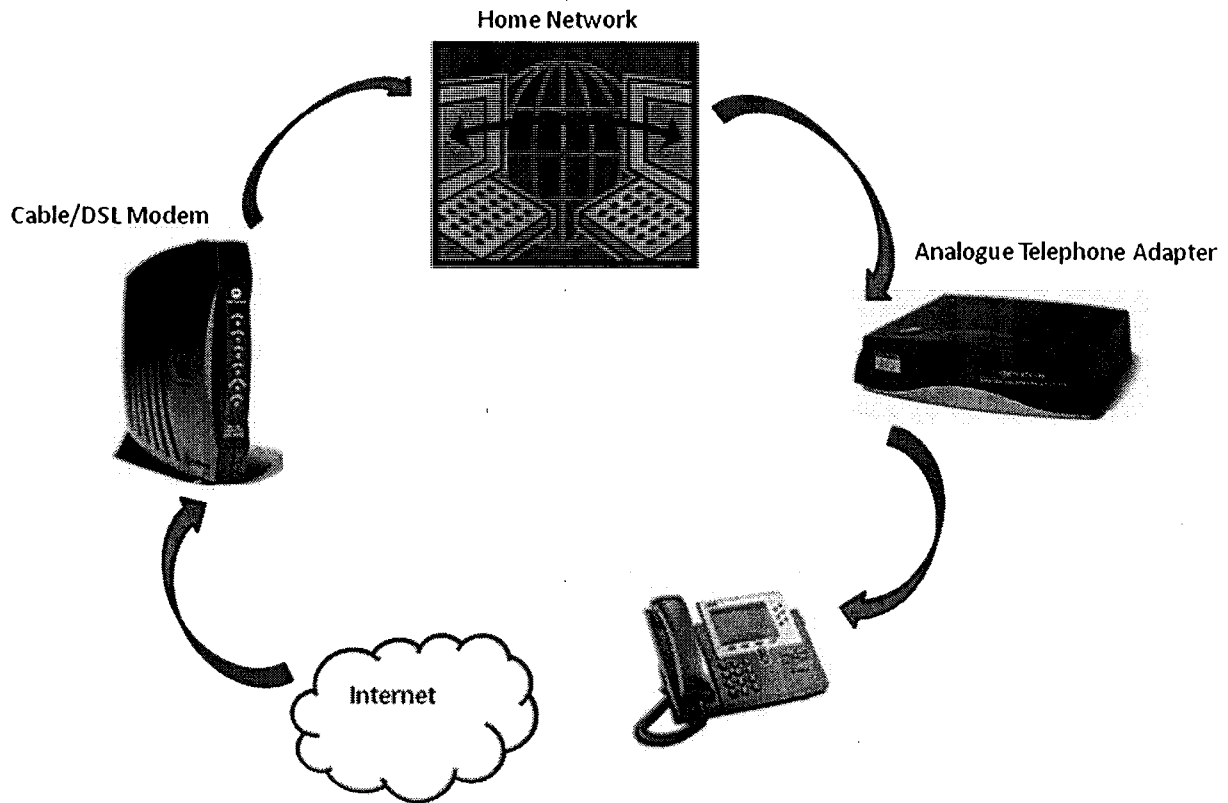


Figure 2.4 Home telephone service using SIP

With the system adopted above, subscribers use their regular PBX phones to make and receive calls. However, they do not incur costs of the regular circuit-switched long-distance providers. The regular telephone set is hooked to a Cisco ATA (Analogue Telephone Adaptor). An ATA interfaces a regular analog telephone to the IP-based network. The Cisco ATA is either connected directly to the internet, or to your existing home network. This system allows VoIP calls to be made from your regular phone set while saving about 50% on calls.

VONAGE™ provides all regular services such as voicemail, caller-id, and call waiting/forwarding/conferencing. Moreover, using the concept of PSTN virtual

CHAPTER 2. BACKGROUND

number discussed earlier, VONAGE™ allows its subscribers to pick one or more area codes for their virtual PBX number.

2. *Instant Messaging (IM)*: Take a peek at your Windows or Linux taskbar. It's almost inescapable that you will find one of these logos:

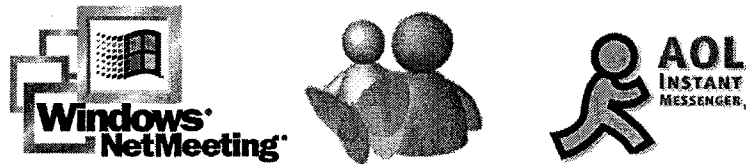


Figure 2.5 IM programs using SIP

Versions of these programs for Windows XP are already implemented using SIP. Instant messaging refers to the transfer of messages between users in near real-time. Messages are usually sent back and forth between two or more parties to give the impression of a real conversation[4].

Instant Messaging has become an integral part of our everyday communication. It's very convenient, effective and mobile. You can log in from any PC, laptop, handheld or even a cell-phone and you immediately retrieve your buddy or contact list along with all your preferences. Instant messaging connects co-workers, friends and families in an easy and simple way. A user can change his status (e.g. appear busy or offline) and have control over who can see him and who can't.

CHAPTER 2. BACKGROUND

3. *Others*: Click-to-talk call center buttons, E-learning solutions and presence-based notification services (e.g. stock quotes) are another three examples of SIP applications.

SMS-to-IM, IM-to-SMS, presence over GPRS and multimedia support in 3G mobile networks are examples of services that would result if SIP is integrated with 3G wireless technology.

SIP entities already exist today in the market. Industry leading businesses such as Cisco™ offers a wide range of SIP servers, gateways and phones. Similarly, Alcatel™ offers a proxy server with a gateway used to automatically establish communication between IP-based devices/networks and the PSTN ones.

SIP supports control of Network Appliances i.e. dedicated function consumer devices containing networked processors[43] [42][41]. Figure 2.6 depicts how SIP can be used to control devices from the outside world. Using this design, a person would be able (using a GUI from his office desktop or wireless PDA) to set his alarm clock, turn off the washing machine, and switch on the heating system before he leaves the office. They also proposed a Device Messaging Protocol (DMP) to be used instead of SDP.

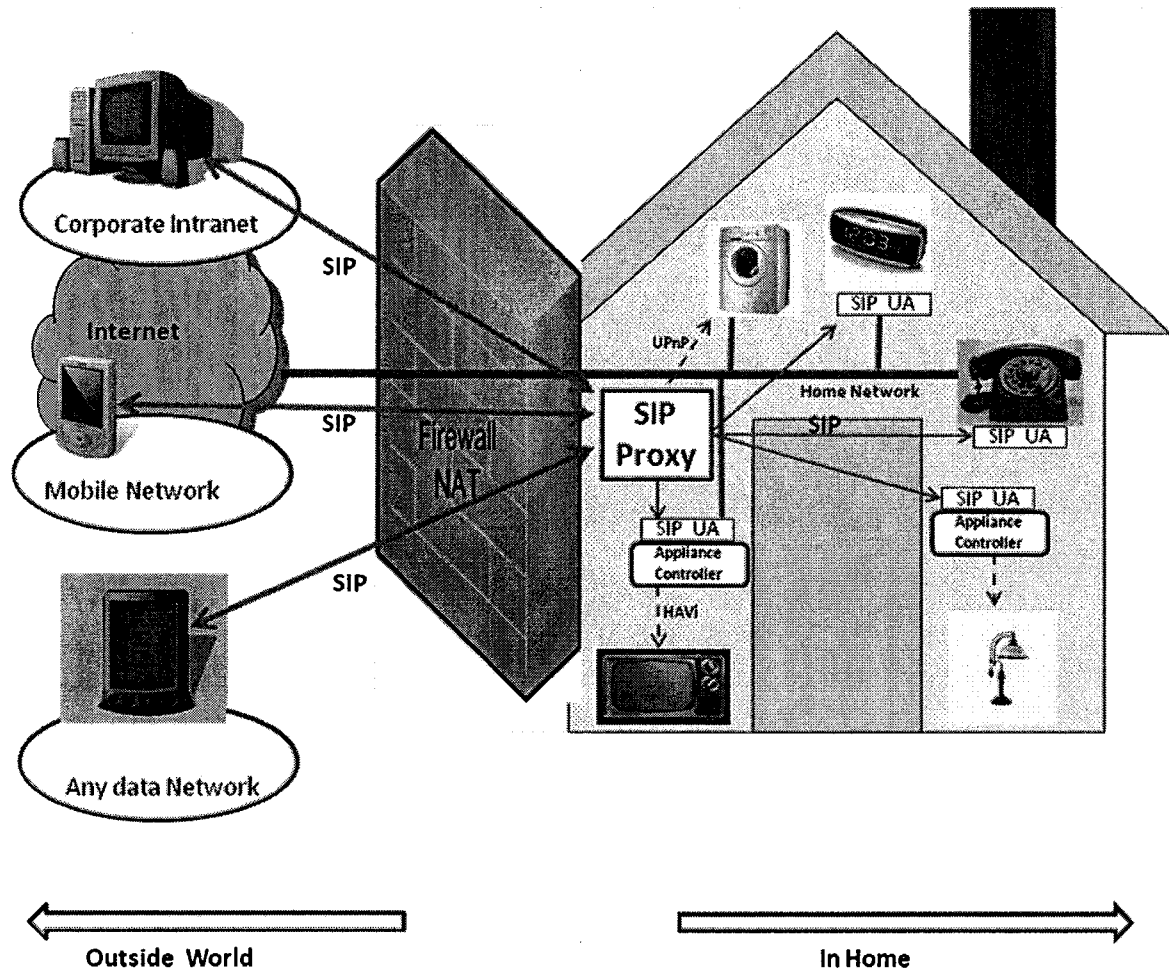


Figure 2.6 Controlling Devices using SIP [43]

2.2 IP Multimedia Subsystem

2.2.1 IMS Overview

IP Multimedia Subsystem (IMS) is a set of specifications that describes the Next Generation Networking (NGN) architecture for implementing IP based telephony and multimedia services.

CHAPTER 2. BACKGROUND

IMS is defined by 3GPP as new subsystem, i.e. a new mobile network infrastructure that enables the convergence of voice, video, data and mobile network technology over an IP-based infrastructure. It was designed to fill the gap between the existing traditional telecommunications technology and Internet technology. This will allow operators to offer new, innovative services that shareholders and end users are expecting. IMS is the key element in the 3G architecture that makes it possible to provide ubiquitous cellular access to all the services that the Internet provides. Imagine yourself accessing your favorite web pages, reading your email, watching a movie, or taking part in a videoconference wherever you are by simply pulling a 3G hand-held device out of your pocket. This is the IMS vision.

2.2.2 IMS Architecture

Before exploring the general architecture in the IMS we should keep in mind that 3GPP [1] does not standardize nodes, but functions. This means that the IMS architecture is a collection of functions linked by standardized interfaces. Implementers are free to combine two functions into a single node (e.g., into a single physical box). Similarly, implementers can split a single function into two or more nodes.

In general, most vendors follow the IMS architecture closely and implement each function into a single node. Still, it is possible to find nodes implementing more than one function and functions distributed over more than one node. Figure 2 depicts an overview of the IMS architecture as implemented by Fraunhofer Institute FOKUS.

CHAPTER 2. BACKGROUND

FOKUS IMS architecture supports a wide range of services that are enabled based on SIP protocols. It delivers multimedia services that can be accessed by a user from various devices via an IP network or traditional telephony system.

The main components of the FOKUS IMS Core Network [19], as listed in Figure 2.7 are:

Home Subscriber Server (HSS): is the central repository for user-related information. Technically, the HSS is an evolution of the HLR (Home Location Register), which is a GSM node. The HSS contains all the user-related subscription data required to handle multimedia sessions. These data include, among other items, location information,

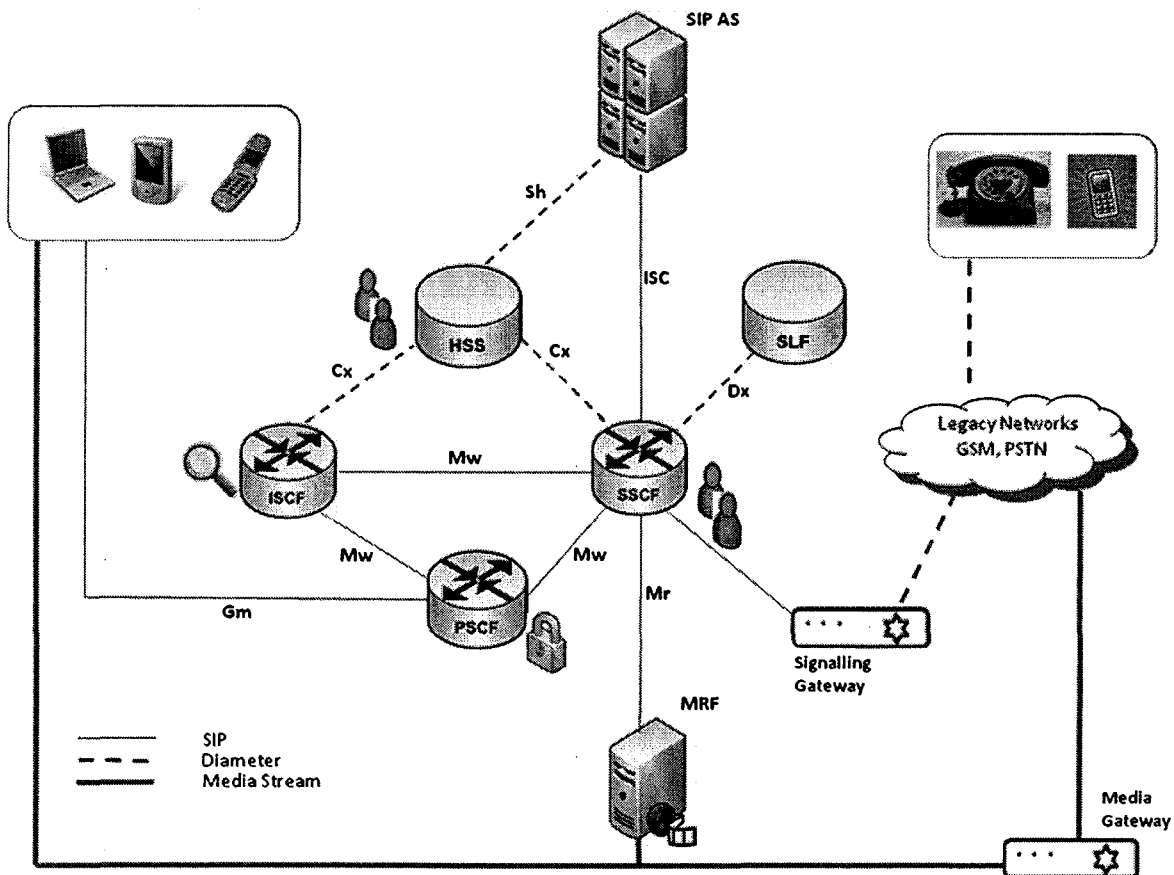


Figure 2.7 IMS Architecture [19]

CHAPTER 2. BACKGROUND

Security information (including both authentication and authorization information), user profile information (including the services that the user has subscribed to), and the S-CSCF (Serving-CSCF) allocated to the user. For example, this is where the public URI and locations are stored and linked together.

A network may contain more than one HSS, in case the number of subscribers is too high to be handled by a single HSS. In any case, all the data related to a particular user are stored in a single HSS. Both the HSS and the SLF communicate through the Diameter protocol RFC 3588 [36]. Networks with a single HSS do not need SLF. On the other hand, networks with more than one HSS do require an SLF. The SLF is a simple database that maps users' addresses to HSSs.

Application Server (AS): It is an Application Server that hosts and executes IP Multimedia Services based on SIP. Depending on the actual service the AS can operate in SIP proxy mode, SIP UA (User Agent) mode (i.e., endpoint), or SIP B2BUA (Back-to-Back User Agent) mode (i.e., a concatenation of two SIP User Agents). AS implements ISC interfaces to exchange information with S-CSCF.

The AS can be located either in the home network or in an external third-party network to which the home operator maintains a service agreement. In any case, if the AS is located outside the home network, it does not interface the HSS.

Media Resource Function (MRF): Provides a source of media in the home network.

CHAPTER 2. BACKGROUND

The MRF provides the home network with the ability to play announcements, mix media streams transcode between different codecs, obtain statistics, and do any sort of media analysis.

The MRF is further divided into a signaling plane node called the MRFC (Media Resource Function Controller) and a media plane node called the MRFP (Media Resource Function Processor). The MRFC acts as a SIP User Agent and contains a SIP interface towards the S-CSCF. The MRFP implements all the media-related functions, such as playing and mixing media.

Call Session Control Function (CSCF): It is a SIP server or proxy used to process SIP signaling packets in the IMS. There are three types of CSCFs, depending on the functionality they provide. All of them are collectively known as CSCFs, but any CSCF belongs to one of the following three categories.

- Proxy-CSCF (**P-CSCF**): The P-CSCF is the first point of contact between the IMS terminal and the IMS network. It acts as a SIP proxy server; this means that all the requests initiated by the IMS terminal or destined for the IMS terminal traverse the P-CSCF. The P-CSCF forwards SIP requests and responses in the appropriate direction towards the IMS terminal or toward the IMS network.

The P-CSCF is allocated to the IMS terminal during IMS registration and does not change for the duration of the registration so IMS terminal communicates with a single P-CSCF during the registration.

CHAPTER 2. BACKGROUND

The P-CSCF authenticates users by establishing an IPsec security association with IMS terminal to prevent spoofing attacks and reply attacks

Once the P-CSCF authenticates the user as part of security association establishment the P-CSCF asserts the identity of the user to the rest of the nodes in the network. This way, other nodes do not need to further authenticate the user, because they trust the P-CSCF. The rest of the nodes in the network user's identity asserted by the P-CSCF have a number of purposes, such as providing personalized services and generating account records.

Additionally, the P-CSCF verifies the correctness of SIP requests sent by the IMS terminal. This verification keeps IMS terminals from creating SIP requests that are not built according to SIP rules.

The P-CSCF also includes a compressor and a decompressor of SIP messages. SIP messages can be large, given that SIP is a text-based protocol. While a SIP message can be transmitted over a broadband connection in a fairly short time, transmitting large SIP messages over a narrowband channel, such as some radio links, may take a few seconds. The mechanism used to reduce the time to transmit a SIP message is to compress the message, send it over the air interface, and decompress it at the other end.

The P-CSCF may include a PDF (Policy Decision Function). The PDF may be integrated with the P-CSCF or be implemented as a stand-alone unit. The PDF authorizes media plane resources and manages Quality of Service over the media plane.

CHAPTER 2. BACKGROUND

The P-CSCF also generates charging information toward a charging collection node. An IMS network usually includes a number of P-CSCFs for the sake of scalability and redundancy. Each P-CSCF serves a number of IMS terminals, depending on the capacity of the node. The P-CSCF may be located either in the visited network or in the home network

- Interrogating-CSCF (**I-CSCF**): It is the entry point to IMS from other networks retrieves user location information and routes the SIP request to the appropriate S-CSCF. It acts as a SIP proxy located at the edge of an administrative domain and implements Diameter Cx to the HSS and Dx interfaces to the SLF. The address of the I-CSCF is listed in the DNS (Domain Name System) records of the domain. When a SIP server follows SIP procedures to find the next SIP hop for a particular message the SIP server obtains the address of an I-CSCF of the destination domain. I-CSCF may optionally encrypt the parts of the SIP messages that contain sensitive information about the domain, such as the number of servers in the domain, their DNS names, or their capacity. I-CSCF located usually at the home network.

- Serving-CSCF (**P-CSCF**): The S-CSCF is the central node of the signaling plane. Located always in the home network, it performs session control as well as SIP server functionality and acts as a SIP registrar so it maintains a binding between the user location (e.g., the IP address of the terminal the user is logged on) and the user's SIP address of record (also known as a Public User Identity).

CHAPTER 2. BACKGROUND

Like the I-CSCF the S-CSCF also implements a Diameter interface to the HSS. The main reasons to interface the HSS are as follows.

- To download the authentication vectors of the user who is trying to access the IMS from the HSS. The S-CSCF uses these vectors to authenticate the user.
- To download the user profile from the HSS. The user profile includes the service profile, which is a set of triggers that may cause a SIP message to be routed through one or more application servers.
- To inform the HSS that this is the S-CSCF allocated to the user for the duration of the registration.

The S-CSCF inspects every SIP message sent or received through the IMS terminal and determines whether the SIP signaling should visit one or more application servers. Those application servers would potentially provide a service to the user.

One of the main functions of the S-CSCF is to provide SIP routing services. If the user dials a telephone number instead of a SIP URI the S-CSCF provides translation services as described in RFC 2916 [37].

The S-CSCF also enforces the policy of the network operator. For example, a user may not be authorized to establish certain types of sessions. The S-CSCF keeps users from performing unauthorized operations.

CHAPTER 2. BACKGROUND

A network usually includes a number of S-CSCFs for the sake of scalability and redundancy. Each S-CSCF serves a number of IMS terminals, depending on the capacity of the node.

2.2.3 IMS benefits

IMS is designed to provide a number of key functionalities required to enable new IP services via mobile networks. This new realm of IP services must take into account the complexity of multimedia, constraints of the underlying network, managing mobility and managing the multitude of emerging applications.

Although IMS was designed for mobile networks, it can also be used to provide services to fixed network at the same time, providing unique mixtures of services with transparency to the end users. Some of the key functionalities are briefly explained below:

1. **Multimedia Session Negotiation and Management- Key to IP**

Communication Services

IMS uses SIP for multimedia session negotiation and session management. IMS is essentially a mobile SIP network designed to support mobility functionality, where IMS provides routing, network location and addressing functionalities.

In contrast to the CS (Circuit-Switched) and PS (Packet-Switched) domains, IMS domain enables any type of media session to be established (e.g. voice, video, text, etc.). It also allows the service creator the ability to combine services from CS

CHAPTER 2. BACKGROUND

and PS domain in the same session, and for sessions to be dynamically modified in the fly (e.g. adding a video component to an existing voice session). This capability opens a number of new and innovative user-to-user and multi-user services such as enhanced voice services, video telephony, chat, push to talk (PoC) and multimedia conferencing, all of which are based on the concept of multimedia session.

2. Mobility Management- Critical for Roaming

The underlying IMS infrastructure enables mobile IP communication services via its ability to find other users in the network and then establish a session with that user. The key IMS components enabling mobility management are CSCF and HSS. The HSS holds all of the key subscriber data and enable users to find and communicate with other users. The CSCF is essentially a proxy, which aids in the setup and management of sessions and forwards messages between IMS networks. IMS is critical to enable service access regardless of the end user's geographical location.

3. Quality of Service (QoS) – Key to Quality Real-time Service Realization

IMS will provide effective and standardized solutions for operators who want to implement real-time IP mobile services without gambling on the best effort transmission and the resulting customer dissatisfaction.

CHAPTER 2. BACKGROUND

Real-time mobile IP communication is difficult due to the fluctuating bandwidths, which severely affect the transmission of IP packets through the network. In normal IP networks, IP transport would be what is known as ‘best effort’ meaning that the network will do its best to ensure the required bandwidths, but there is no guarantee. The result is that real-time mobile IP services function or not at all (i.e. voice quality is poor or garbled, video jitter, etc) depending on the bandwidth availability and network congestion.

The Quality of Service (QoS) mechanisms were developed in order to overcome these issues and provide some types of guaranteed level of transmission instead of ‘best effort’. QoS ensures the critical elements of IP transmission such as transmission rate, gateway delay and error rates can be measured, improved and guaranteed in advance. Users are able to specify the level of quality they require depending on the type of the service and users’ circumstances.

The intelligence required to enable QoS within a mobile IP network is specified in IMS in the form of an entity known as the Policy Decision Function (PDF). The PDF interacts with and controls the underlying packet network.

4. Service Execution, Control and Integration – Foundation for a Robust

Service Platform

In complex mobile service landscape wherein the operator has deployed a large number of services, it is crucial that the operator is able to control the invocation

CHAPTER 2. BACKGROUND

of services and interaction between the various services components. In CS and PS domains, service execution is application-controlled, which makes service integration increasingly complex and reduces overall service transparency and control. IMS meets this change by providing efficient service provisioning functionality.

When a user registers on the IMS network, his SSP (Subscriber User Profile) is downloaded by the CSCF from the HSS. The information contained within the SSP allows the CSCF to:

- Which services need to be executed
- Determine the order of services execution
- Determine the addresses of the application servers that should execute the requested end-user services

IMS infrastructure effectively control and manage complexities involved in filtering, triggering and integration of services.

5. 3rd Party Developer Interfaces

IMS provides the standardized architecture for enabling advanced IP service deployment. Varieties of IMS services can be developed independently and at the same time utilize the common features of the IMS infrastructure. This will facilitate service integration, as well as interoperability between mobile and fixed networks. Additionally, roaming functionality is automatically supported within little or no additional effort.

CHAPTER 2. BACKGROUND

2.2.4 IMS Applications

The mobile industry is in a transition phase from traditional voice and short message service centric business to a variety of new and exciting multimedia services and applications. IMS enables new services between mobile and fixed devices. Examples of such services are described below:

1. **Push to Talk over Cellular (PoC)** is a walkie-talkie type of service, unlike regular voice calls, which are full-duplex, PoC is a half-duplex service; that is, only one user can speak at a time. PoC uses cellular access and radio network resources very efficiently. Networks resources are thereby used only one-way for the duration of talk spurts instead of two-way for an entire call session. Users can communicate in both one-to-one and one-to-many fashions, with short set-up times.
2. **Real Time Video Sharing** is a peer-to-peer, multimedia streaming service that can be offered entirely as a PS service or as a “combinational” service, combining the capabilities of the CS and IMS packet switched domains.

In a combinational scenario, the service enriches the user experience during a circuit switched telephony call by exchanging pictures, video clips or live video over a simultaneous IMS packet-switched connection.

In both IMS combinational and IMS packet switched scenarios, the media is delivered and consumed almost real-time.

CHAPTER 2. BACKGROUND

- 3. Interactive Gaming** mobile users worldwide downloaded thousands of millions of Java games in the past years.

Taking the end-user demand for basic gaming with mobile terminals into consideration, it is difficult to foresee an even faster service take off with the capability to establish interactive gaming sessions between players.

- 4. Shared Folders** Content sharing enables users to share files between terminals. A typical use case includes sharing of files or contents such as images, documents, notes, contacts or calendar information, even simultaneously while in a voice-call session.

- 5. Instant Messaging Services** is a communication service that allows end-users to send and receive messages instantly. Instant Messaging is well known in today's Internet community.

IMS will bring the same service experience to the mobile world, including interoperability without the need for legacy infrastructure. Messages can contain any MIME type media content such as text, image, audio or video clips, application data or a combination of these. The message is sent through the packet data network to the IMS, which locates the terminating IP client and routes the message to the recipient.

An Instant Messaging service can offer a store-and-forward (S&F) function – if message cannot be delivered to the receiver straight away, it can be stored in an S&F element in IMS, which will forward the message to the recipients when they become available. S&F functionality can also use presence information to

CHAPTER 2. BACKGROUND

schedule message forwarding. Instant message can also be delivered to non-IMS terminal.

- 6. Voice Messaging** is a form of instant messaging where the content of the message is an audio file. Using an application in the terminal, users can record the message instantly or use existing audio files stored in the terminal's folders. Voice messages can be sent to one or many recipients.

Voice messaging is a quicker and easier way of messaging compared to writing short messages with a phone keyboard. Voice is also much more natural and personal way of communication than written text. Personal voice or even a song enriches messaging and gives recipients a more personalized experience.

- 7. IMS enabled Voice and Video Telephony** IMS calls are carried over a packet core network (VoIP). Video Telephony is seen as a critical end-user service in Mobile Networks. SIP enables Voice and Video Telephony Person-to-Person and Multiparty sessions over an IP network. The issues around VoIP and Video Telephony calls are around QoS in packet core networks, as well as on interoperability with the Public Switched Telephone Network (PSTN) and legacy phones, and interworking with existing domains for Video Telephony.

The IMS infrastructure provides several features to manage and overcome these issues, IMS users are able to specify the level of QoS they require depending on the type of the service and users' circumstances. IMS terminals are able to

CHAPTER 2. BACKGROUND

connect to both CS and PS networks, so, when a user wants to call a phone in the PSTN or a cellular phone the IMS terminal chooses to use the CS domain.

- 8. Video Conferencing** IMS Video-conferencing service extends the point-to-point video call to a multi-point service. Video-conferencing requires an IMS conference bridge service, which links the multiple point-to-point video calls together, and implements the associated service logic. The video telephony connections are made point-to-point from the terminals to the conference bridge, which takes care of joining the point-to-point connections into a conference. The conference bridge is not concerned about the underlying infrastructure and client devices and assumes that audio- and video connections are provided by the appropriate standard and that these connections are delivered over the IP network.

2.2.5 IMS Message Flow

Registering over the IMS Network is easy to understand with the SIP protocol. As we see in Figure 2.8 Alice sends a REGISTER request to the P-CSCF which acts as gateway to the IMS network where every message from outside is analyzed and routed. The REGISTER request contains the realm Alice wants to register to. The **To** header field contains Alice's public URI and the **Contact** header field contains Alice's username and current location. The current location is important as it specifies where the answer to the REGISTER request will be sent. Once the registration is accepted, Alice receives a 200 OK message.

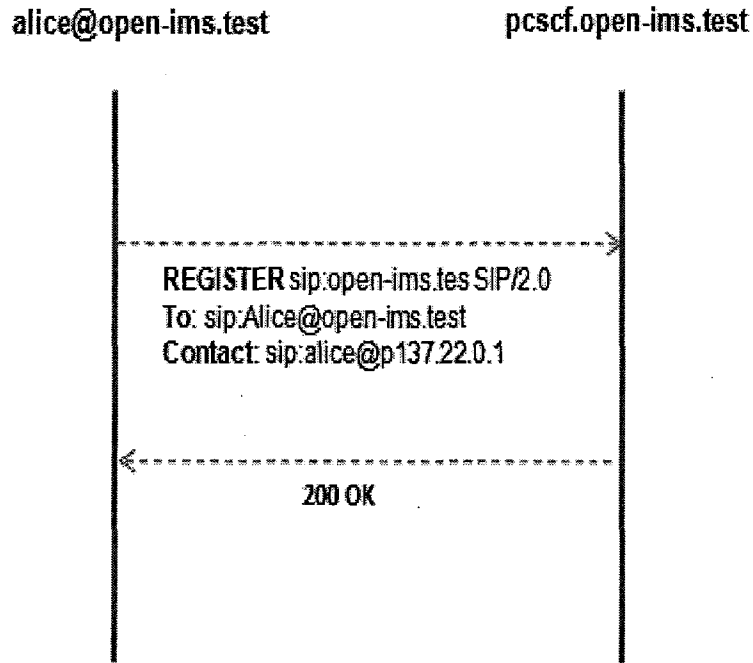


Figure 2.8 Example of Registration to IMS

The P-CSCF does not directly handle the registration request inside the IMS rather it forwards the request to the I-CSCF which sends the registration info to the user database (HSS). The I-CSCF checks if there is a S-CSCF allocated to Alice, if there is one allocated, the I-CSCF transfers the request to the S-CSCF which is the proxy actually handling the REGISTER request. The S-CSCF contacts the HSS using the Diameter protocol to check if Alice has an account that is not already in use and sends its answer back to the I-CSCF, The I-CSCF forwards the message to the P-CSCF and finally to Alice.

CHAPTER 2. BACKGROUND

2.3 Summary

This chapter should have provided background knowledge related to the thesis. First, the SIP protocol is discussed in detail; this is followed by IMS descriptions. From the topics discussed in this chapter, it can be concluded that SIP and IMS provide fundamental tools and techniques for designing the proposed architecture presented in this thesis.

The next chapter discusses session mobility. To succeed in that matter, we exhaust much needed discussion to provide adequate information about session mobility's main techniques, transfer modes, types of transferred media and the requirements for seamless handoff in session mobility.

Chapter 3 Mobility Management

In this chapter, we delve into some issues regarding mobility management; we first give a brief overview of mobility, and then introduce SIP as a highly-suitable candidate for mobility management. In this section we discuss types of mobility including service, terminal and personal mobility.

We then study session mobility, outlining its main techniques, transfer modes, types of transferred media and the requirements for seamless handoff in session mobility. Finally, we review some of the most important related work

3.1 Overview

The explosive growth of internet and increasing demand for ubiquitous mobile wireless services encourages design activities in all IP wireless networks that provide integrated data, voice and multimedia services for roaming users [2].

As mobile devices improve and include more enhanced capabilities for IP-based multimedia communications, they will remain limited in term of bandwidth, display size and computational power. Stationary IP multimedia endpoints, including hardware IP phones, videoconferencing units, embedded devices and software phones allow more

CHAPTER 3. MOBILITY MANAGEMENT

convenience of use, but are not mobile. Moving active multimedia sessions between these devices allows mobile and stationary devices to be used concurrently or interchangeably in mid-session, combining their advantages into a single virtual device.

Mobility management is an important issue for a mobile device; therefore several forums have selected SIP for session and mobility management[13] [11]. This is mostly based on strengths of SIP such as simplicity, scalability, extensibility and modularity. SIP is especially used for mobility support because it possesses the following advantages [34][22],

- SIP allows user roaming depending on appliances rather than networks
- By means of SIP address binding and registration, SIP provides route optimization and improves real-time services
- SIP supports mobility at level above IP terminal

SIP supports different type of mobility layers which include terminal, service, personal and session mobility. It has the capability to provide an approach to support mobility on application layer and is completely independent of lower layers. In SIP, a user needs to register to SIP registrar whenever his/her address changes. If a user registers several addresses, SIP proxy forks SIP INVITE to these addresses sequentially or parallel until the call has been set up. The mechanism of binding users with their address is the foundation of SIP mobility support. For example in the Figure 3.1 shown below, a user bob migrates to foreign network, he needs to register his new address to his home proxy. When Alice calls Bob, Bob's home proxy will forward Alice's request to his current address at a foreign network, this way Alice's call can always reach mobile user Bob.

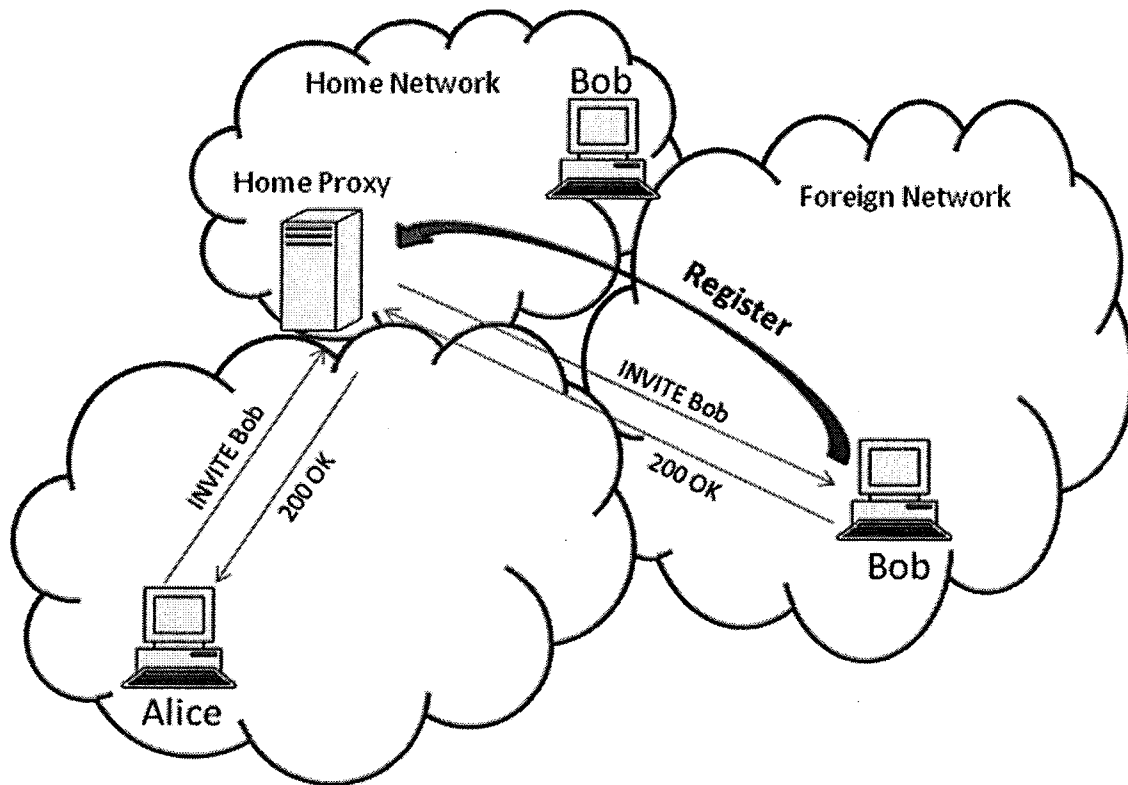


Figure 3.1 SIP support Mobility

3.2 Types of mobility

3.2.1 Personal Mobility

Personal Mobility refers to the user's ability to access mobility services from anywhere, at anytime, using any terminal [17]. In other words, personal mobility allows the addressing of a single user, located at different terminals, by the same logical address. SIP provides transparent support of name mapping and redirection services through the use of the SIP Registrar and Location Services Server. These components keep track of all the possible terminals associated with a user, and the permanent and temporary addresses of each of those terminals, at any given time. This provides SIP with inherent personal mobility support users can use a single personal identifier in all occasions,

CHAPTER 3. MOBILITY MANAGEMENT

regardless of the terminal(s) used or their network locations. Any incoming messages or calls will be redirected to the appropriate active terminal(s).

In the case where a single user is associated with multiple terminals, SIP can provide personal mobility support through the use of a SIP Registrar and Location Services Server, and a simple call forking procedure. The actual address of the active terminal is determined dynamically, as a session initiation request is received. This disassociates the user from any one particular terminal or network location, providing true personal mobility.

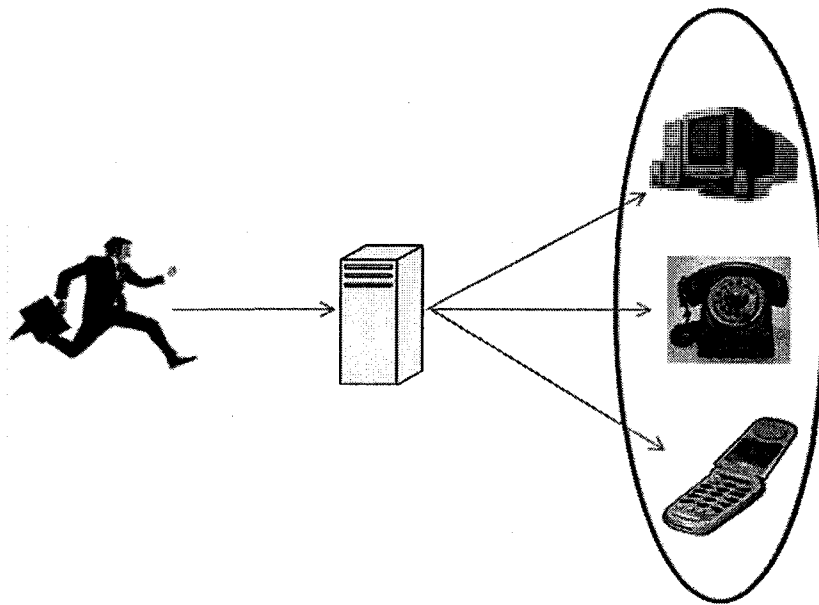


Figure 3.2 Personal Mobility

CHAPTER 3. MOBILITY MANAGEMENT

3.2.2 Terminal Mobility

Terminal Mobility refers to the user's ability to use his/her terminal to move across heterogeneous networks while having access to the same set of subscribed services. SIP similar to Personal Mobility, Terminal Mobility can be provided in SIP through the use of the SIP Registrar and Redirect Server. As the terminal moves across heterogeneous networks new temporary identifiers IP addresses are assigned to the terminal. These are updated with the SIP Registrar by using the REGISTER method. The current location of the device is always up-to-date so that messages can be redirected successfully, in a time-efficient manner. As we see SIP provides transparent support of Terminal Mobility using registration and session update mechanism. This approach fails to support dual movement problem where two communication peers move at the same time. And the correspondent must get involved in every movement of the communication peer.

To solve above problems, some hybrid mobility management schemes are proposed where use SIP for UDP-based applications and Mobile IP [6] for TCP-based applications [27][7][33]. And some pure SIP schemes are proposed where use SIP Back-to-Back User Agent (B2BUA) like mobility agent in Mobile IP [3][33][10].

3.2.3 Service Mobility

Service mobility allows users to maintain access to their services even while moving or changing devices and network services provider. For example, services that users will likely want to maintain include their phone books, call logs, buddy list, media preferences and incoming call handling instructions.

CHAPTER 3. MOBILITY MANAGEMENT

One solution for service mobility is to have the user carry information with him, either a SIM card or a memory disk or a PDA. With respect to service mobility two aspects must be considered: maintaining adequate QoS for the duration of a session regardless of the network changes, and ensuring that the users have access to all their subscribed services regardless of the point of attachment to the network [12].

3.3 Session Mobility

Session Mobility refers to the user's ability to maintain an active session while switching between terminals. SIP can successfully implement session mobility through the use of the re-INVITE method. This is an INVITE method sent while a multimedia session is in progress. While maintaining the existing session alive, new terminals can be added to the session and existing ones can be removed. Changes could be made to the parameters of the session in progress in order to match the capabilities of the newly added terminals.

3.3.1 Techniques

We have two options of transfer in session mobility as specified below:

- 1. Transfer and Retrieval*

Transfer means to move the active session from the current device to one or more other devices. Retrieval means to transfer a session currently on a remote device back to the local device. For example, a user in videoconference communication with his handheld device enters new location where more adapted video display/acquisition devices are available. In this case, the user can transfer a video

CHAPTER 3. MOBILITY MANAGEMENT

session to these devices. Before walking away, he can retrieve the video stream to his mobile device for continued communication.

2. *Whole and split transfer*

Session media may either be transferred completely to a single device or be split across multiple devices. For instance, a user may only wish to transfer the video portion of his session while maintaining the audio portion on his Personal Digital Assistant (PDA). Alternatively, he may find separate video and audio devices and may wish to transfer one media service to each of them. Furthermore, even the two directions of a full-duplex session may be split across devices. For example, a PDA's display may be too small for a good view of the other call participant, so the user may transfer video output to a projector and continue to use the PDA camera.

3.3.2 Transfer modes

Two different modes are possible for session transfer:

1. *Mobile Node Control mode*

In Mobile Node Control mode, a signaling session dialog is established with each device used in the transfer. The Mobile Node (MN) updates its session with the Correspondent Node (CN) using Session Description Protocol (SDP) parameters to establish media sessions between the CN and each device, consequently replacing the current media session with the CN. The shortcoming of this approach is that it requires the MN to remain active to maintain sessions. In the Mobile Node Control, the Mobile Node uses third party call control 3PCC [26].

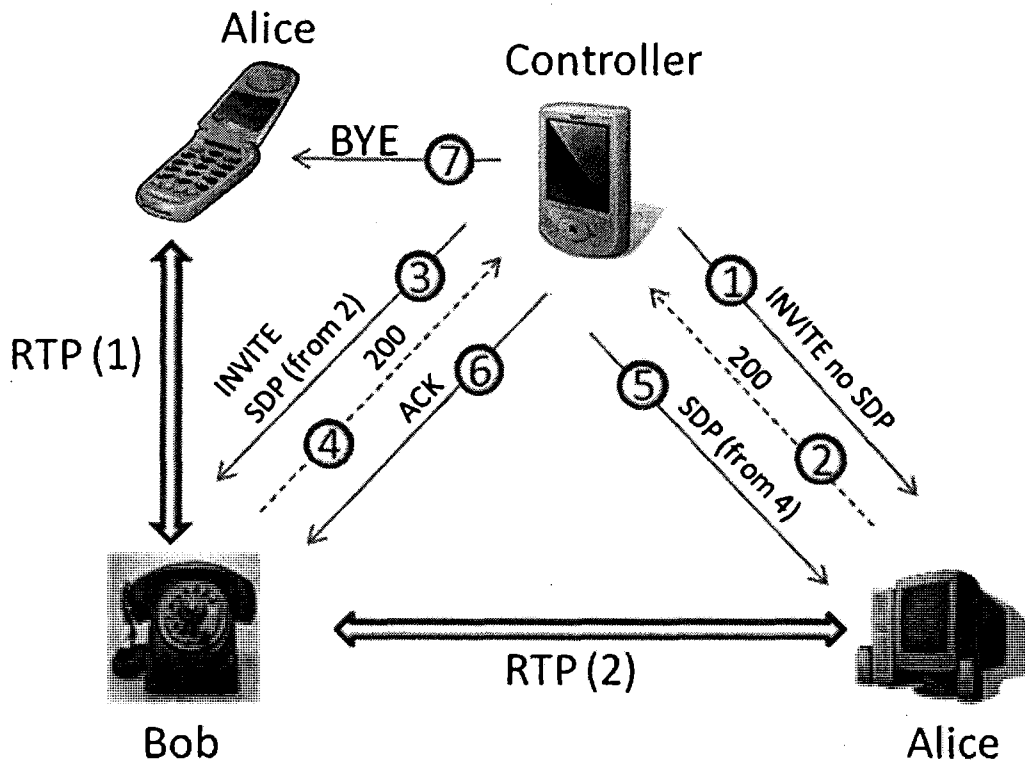


Figure 3.3 Third party call control for session mobility [31]

The mechanism of third party call control for session mobility is illustrated in Figure 3.3. Third party call control allows one entity (controller) to set up and manage a communications relationship between two or more other parties.

In the beginning, Alice used the controller to make a call between Alice's cellular phone and Bob's phone where they communicate through RTP (1). During the phone call Alice wants to continue the call with Bob from her desktop. She commanded the controller to setup a new session between Alice's desktop and Bob's phone (Step 1-6), If the new session setup successfully, Alice's controller

CHAPTER 3. MOBILITY MANAGEMENT

will send a BYE request to terminate the origin session RTP (1). Otherwise, Alice keeps talking to Bob by using her cellular phone.

2. *Session Handoff mode*

A user may need to transfer a session completely because the battery on his mobile device is running out. Alternatively, the user of a stationary device that leaves the area and wishes to transfer the session to his mobile device, will not want the session control to remain on the stationary device when he is away. This could allow others to easily tamper with his call. In such case, Session Handoff mode, which completely transfers the session signaling and media to another device, is useful.

Session handoff mode uses SIP REFER method [40], which is mainly used in applications to transfer calls as illustrated in Figure 3.4 and to handle session mobility. REFER method is used to request a user agent to contact a resource this resource is identified by a URI, which may or may not be a SIP URI. The contact information is included in a new defined header field “Refer-To” which only appears in REFER request, The REFER mechanism includes an implicit subscription to the result of the operation initiated by the recipient of the REFER to keep track on the progress of the refer event.

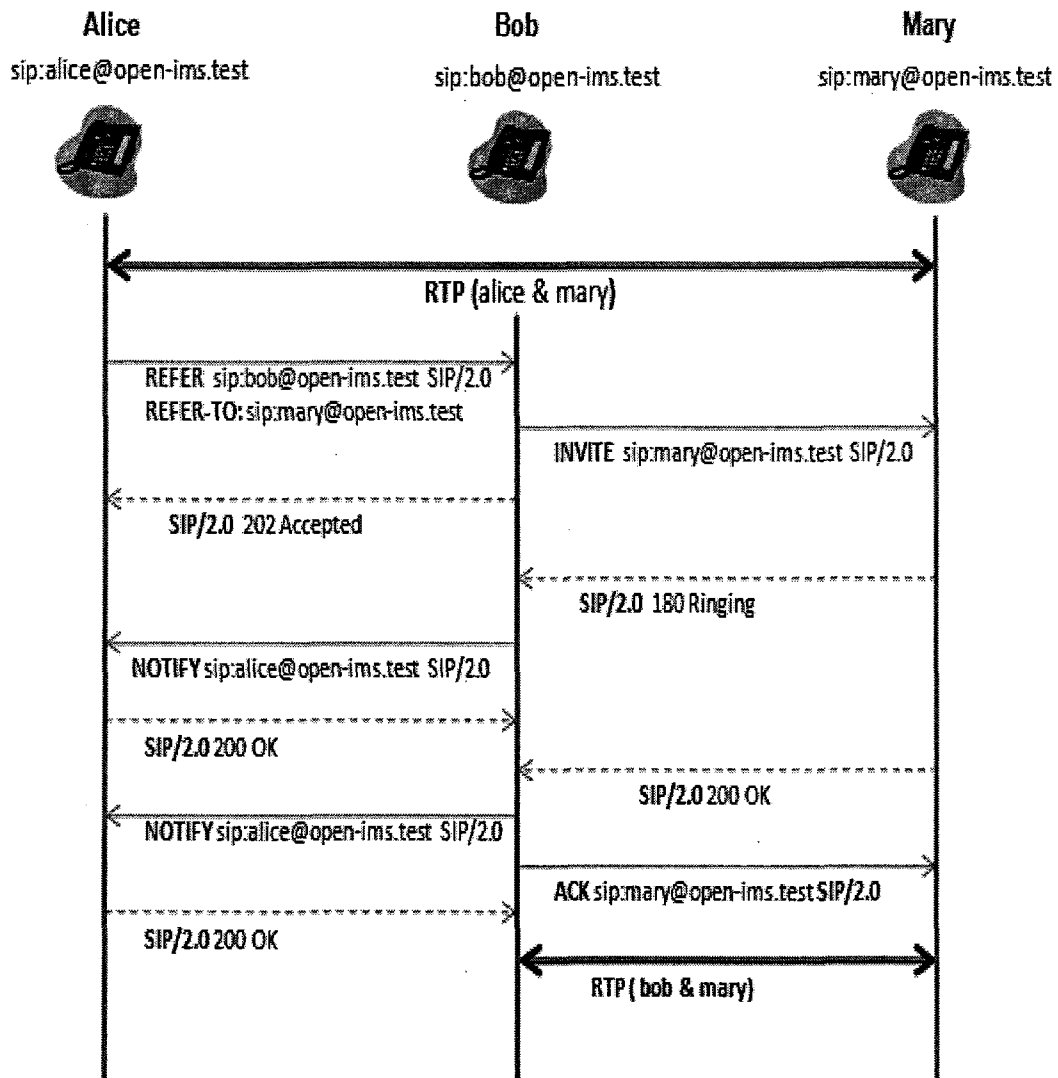


Figure 3.4 SIP REFER Method for Session Mobility

3.3.3 Types of transferred media

A communication session may consist of a number of media types, and a user should be able to transfer any of them to his chosen device. Audio and video are carried by standardized protocols like Real Time Protocol (RTP) [16] and negotiated in the body part of the signaling messages and encoded in some popular format like SDP[32] . Any

CHAPTER 3. MOBILITY MANAGEMENT

examples given for audio and video will work identically for text, as only the payloads differ.

3.3.4 Seamless handoff requirements

To achieve seamless handoff in session mobility some requirements need to be fulfilled as specified below:

1. *Minimum Media Disruption*: The call/video session should appear as not discontinued and the transfer of the call/video from one device to another should be transparent to the user
2. *Instant Media Transfer*: decrease transfer delay which include the time for application setup and media buffering
3. *Device Discovery*: a user is aware of the devices which are available in his local area, along with their capabilities (CPU power, memory available, screen resolution, codecs supported...)
4. *Media Adaptation*: Differences in device capabilities (display size, buffer capacity, media codec ...) should be reconciled

CHAPTER 3. MOBILITY MANAGEMENT

3.4 Related Work

As we stated in the previous section, session mobility can be defined as the capability that allows a user to transfer an ongoing communication session from one device to another. There are many research studies to provide session mobility.

The research work presented in [29][28] supports limited session mobility functionality for specific application, they use session state save and retrieve mechanism. For example, in [29] the authors proposed architecture to support and manage application session transfers based on the MPEG-21 multimedia framework. In [28] the authors proposed an adaptive terminal middleware that performs policy-based dynamic resource selection and host-based session management to hide session failures and resource changes from applications and a user.

Another proposed solution for Session mobility was introduced in [5] by using Session Management Server, this technique uses event publication, subscription and notification mechanism, where mobile node suspended session state to session server management using PUBLISH method and requests the suspended session using the SUBSCRIBE method.

A Third Party Call Control (3PCC) mechanism defined in RFC 3725 [26] was introduced for session mobility. 3PCC allows a device, which may not be a media session sender or a media session receive, to setup and manage a communications relationship between two or more other devices. This solution has one major drawback, since the original session

CHAPTER 3. MOBILITY MANAGEMENT

participant has to remain involved in the session, as it will be contacted to change or terminate the session.

The research work presented in [31], allows a user to continue to communicate with a remote party while changing terminals over multiple devices. User can split the session over multiple devices using SIP mobility header and terminates all the session one time using association records.

Another framework was proposed in [30], to allow user to transfer, split and retrieve session over multiple devices, three new agents have been introduced based on the definition of user agent proposed in RFC 3261. There are session manager, session user, and free node.

In [46] the authors also proposed middleware architecture to resolve the session handoff issue. This work tries to minimize application level support for session transfer mechanisms.

The research work in[45] describes a solution to support session mobility options in the full mesh conferencing by extending existing abstract message protocol and to map them using Session Initiation Protocol. This solution also tries to manage the extra message traffic caused by introducing session mobility in full-mesh depending on the used mode (Mobile Node Control Mode or Session Handoff Mode).

CHAPTER 3. MOBILITY MANAGEMENT

Most of the above solutions cause a segment of media to be empty and make the user feel a lapse in the communication during the handoff time. However, contrary to causing a segment of media to be empty, some solutions make a segment of media redundant from the time that the new media session was set up successfully until the old media session was terminated.

Our work differs from those mentioned work due to the fact that we introduce a complete framework to transfer video session seamlessly without causing a segment of media to be empty or redundant during the handoff time. Our framework is built on top of an IMS network, which we believe to be the best platform for session mobility deployment, as it merges mobility and Internet services. The framework combines two techniques to achieve seamless video handoff: It uses the SIP REFER method, with a proposed Prediction Handoff Manager that predicts the required handoff time to transfer the video session completely.

CHAPTER 3. MOBILITY MANAGEMENT

3.5 Summary

In this chapter, we explored some issues regarding mobility management. Starting by an overview of mobility types including service, terminal and personal mobility, we then proceeded to discuss session mobility, where we outlined their main techniques, transfer modes, types of transferred media and the requirements for seamless handoff in session mobility. Finally, we reviewed some of the related work in session mobility.

In the next chapter, we introduce the proposed architecture of our SIP-based handoff prediction manager; in that course, we provide a detailed description of the elements and clients involved in this system.

Chapter 4 Architecture and Features

In fulfilling our goal of presenting the system's architecture, we have decided to start with the architecture's overview first, in order to provide the reader with solid ground to stand on before throwing more details. The latter come in the shape of a detailed description of the system's elements and clients involved. To demonstrate the functionality of our system, we present a Handoff scenario, the call flow of session transfer and the behavior of the each element in the system is also explained.

4.1 Introduction

As can be inferred from its name, Session Mobility refers to the user's ability to maintain an active session while switching between terminals. Therefore, providing seamless handoff in session mobility is a vital, and also one of the most challenging, issues in the heterogeneous next generation All-IP networks. Most of the existing solutions cause a segment of media to be empty and make the user feel a lapse in the communication during the handoff time. However, contrary to causing a segment of media to be empty, some solutions make a segment of media redundant from the time that the new media session was set up successfully until the old media session was terminated.

4.2 Architecture Overview

Our proposed framework is designed to provide seamless video handoff in session mobility over IP Multimedia Subsystem network. As we mentioned earlier, our framework is built on top of IMS, and employs SIP as a signaling protocol. It combines two techniques to achieve seamless video handoff; It uses of SIP REFER method to transfer the SIP session with a Handoff Prediction Manager (HPM) to predict the required handoff time to transfer the media session between terminals.

Figure 4.1, shows the architecture of our system, which features 3 elements:

1. *Mobile Node (MN)*: a device that has capabilities to handle session mobility. At anytime, the MN can move its active sessions to another available Recipient Node (RN). MN in our system represents the user carrying his own laptop, PDA or smart phone.
2. *Correspondent Node (CN)*: a basic multimedia device participating in a session with the Mobile Node (MN). CN represents the streaming video server in our system.
3. *Recipient Node (RN)*: can be a basic or session mobility-enabled device to preserve system interoperability and compatibility. RN represents the user's Desktop computer in our system.

CHAPTER 4. ARCHITECTURE AND FEATURE

The behavior of each element will be explained in the next sections in addition to the interaction between these elements

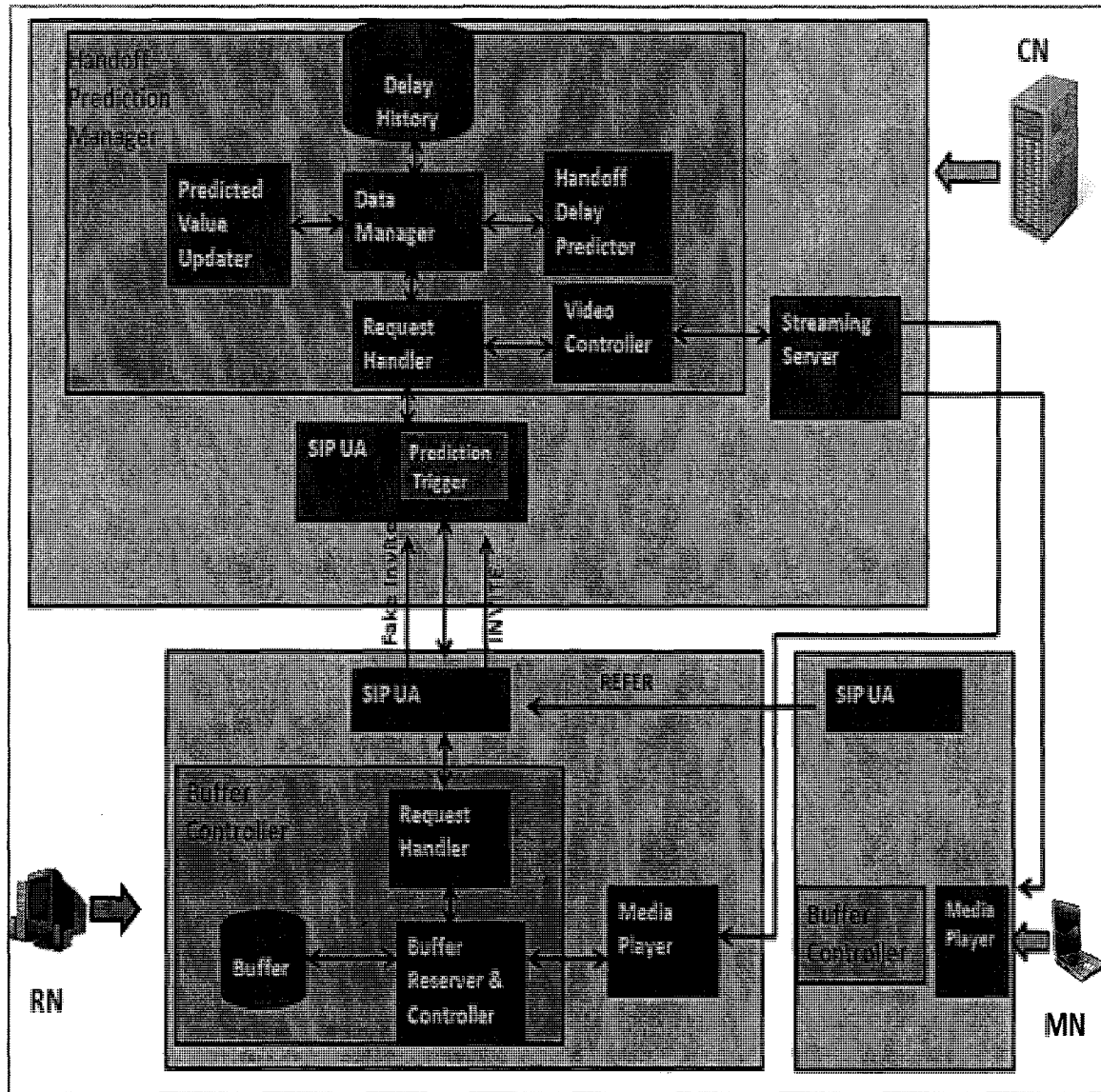


Figure 4.1 Architecture workflow

4.3 System Clients Description

4.3.1 Server Handoff Prediction Manager

CHAPTER 4. ARCHITECTURE AND FEATURE

The framework consists of two clients: Handoff Prediction Manager (HPM) and Buffer Controller. HPM will be installed on the CN and is used to predict the handoff time required to transfer the session from MN to RN. It includes the following components:

Request Handler: handles all requests initiated by SIP user agent, and retrieves frame size and frame rate through Video Controller. It also triggers Data Manager to calculate predicted handoff time.

Video Controller: communicates directly with the streaming video server and passes video parameters, which include frame rate and frame size to Request Handler.

Data Manager: handles requests initiated by Request Handler, and triggers Handoff Delay Predictor and Predicted Value Updater (explained below). Also updates user actual handoff delay history after each session transfer. In case of multiple users, it controls the concurrent access to the handoff Delay History.

Predicted Value Updater: calculates the actual handoff time by recording the timestamp of the INVITE message sent from the RN to the CN (to establish the new session) and the timestamp of the of the BYE message sent from CN to MN (to terminate the old session).

Delay History: communicates directly with Data manager to save the actual handoff time for each user.

CHAPTER 4. ARCHITECTURE AND FEATURE

Handoff Delay Predictor: triggered through request sent by Data Manager to predict the handoff time using specific prediction formula.

4.3.2 Client Buffer Controller

The second client of our proposed framework is the Buffer Controller, which is installed at the RN. It reserves and manages the RN buffer size. Once the RN buffer size reaches a specific threshold; it causes the Buffer Controller to send a message to activate the request handler to trigger the SIP UA. This message indicates that RN is ready to start displaying the cached video.

4.4 Handoff Scenario

The simple scenario presented in Figure 4.2 explains how to transfer a video session between two terminals. Initially the user is watching the video on his PDA while he is walking back home, the video is streamed to the PDA from the video streaming server, once the user enters the house and approaches his desktop computer, the user is interested to continue watching the video on the desktop computer instead of the PDA, so the user presses a button on his PDA to start the transfer of the video session from the PDA to the desktop, in this case if the desktop is capable to play the same video, the old video session to the PDA will stop and a new video session will start on the desktop. The main goal here is to make the video transfer process transparent to the user, so the user does not feel and gap during the handoff process.

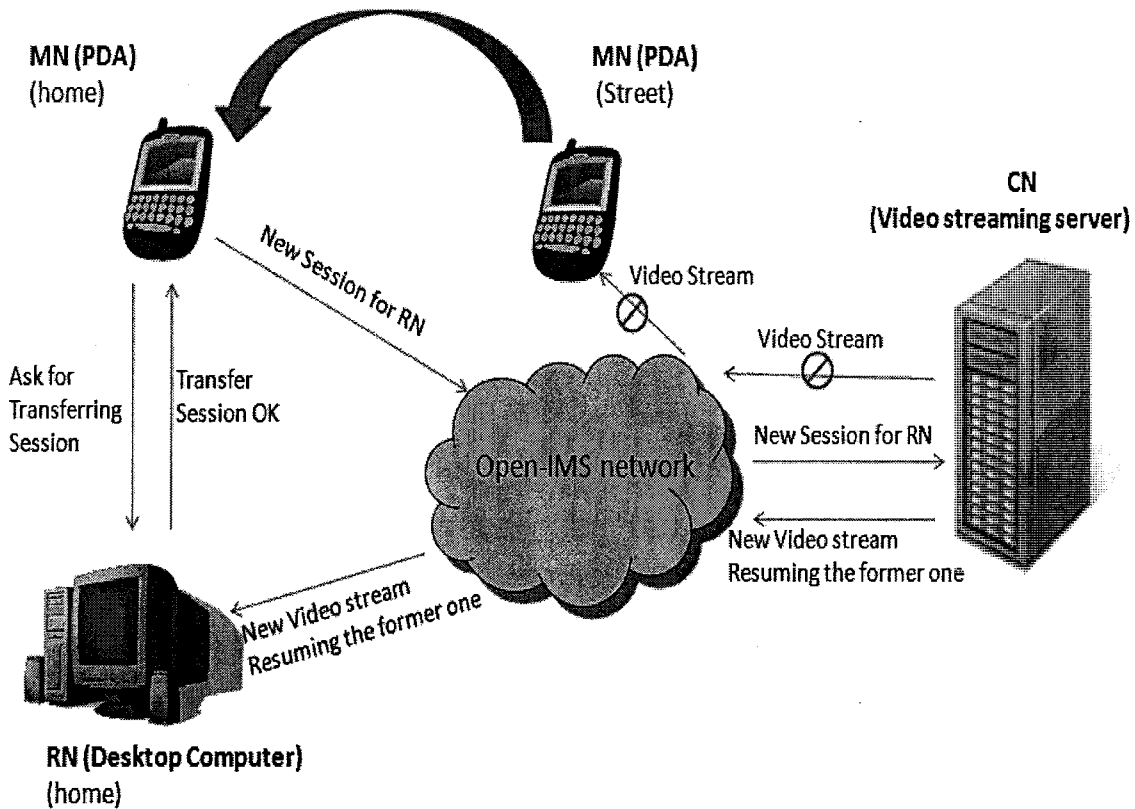


Figure 4.2 Video Transfer Scenario

4.5 System Elements Description

4.5.1 Call Flow of Session Transfer

Figure 4.3 describes the flow of our proposed framework. Initially, the SIP user agent running on the MN registers to the open IMS network by sending a SIP REGISTER request to map the public Uniform Resource Identifier (URI) with the current location of the MN. As soon as the MN is authenticated and identified by the IMS network, it will

CHAPTER 4. ARCHITECTURE AND FEATURE

initiate a SIP INVITE request to establish a new video session with the CN. After the session is established, the CN starts to stream video to the MN.

The MN is aware of the devices which are available in its local area. Once the MN approaches the RN, it will trigger the SIP UA (running on the MN) to send a REFER request to the RN to start transferring the session. If RN supports the media type assigned by the media parameter sent by the REFER request, it will send 202 accepted response to MN and generate an appropriate SDP description for this type of media, then put it into the body of a new INVITE request and send it to the CN; otherwise, RN will send a reject response to MN and reject the REFER request.

The INVITE request sent from RN to CN triggers Handoff Prediction Manager (HPM) in order to predict the number of video frames to be streamed to MN and RN during the handoff process. HPM will return the frame position from which the next transmission to the RN will take off from after the handoff is complete. The frame position is given by the sum of the current frame number (#) sent to the MN and the number of frames (Δ) that will be displayed in the MN during the handoff process; (Δ) can be calculated experimentally [44] as

CHAPTER 4. ARCHITECTURE AND FEATURE

$$\Delta = (\mu + k \cdot \sigma^2) \cdot F$$

μ : Mean of handoff delays

σ^2 : Variance of handoff delays

k : Constant

F : Frame rate

During the handoff process, the user watches the video at the MN only. Simultaneously, CN buffers the video to the RN, rendering the handoff process seamless and transparent to the user. By using the REFER method, combined with the PHM, the video session will have been transferred without any disruption or frame loss. In addition to that, our proposed framework has a recovery algorithm in case (Δ) is overestimated; in that case CN will delay sending the BYE message to MN and OK message to PC until the expected handoff trigger equals ($\# + \Delta$).

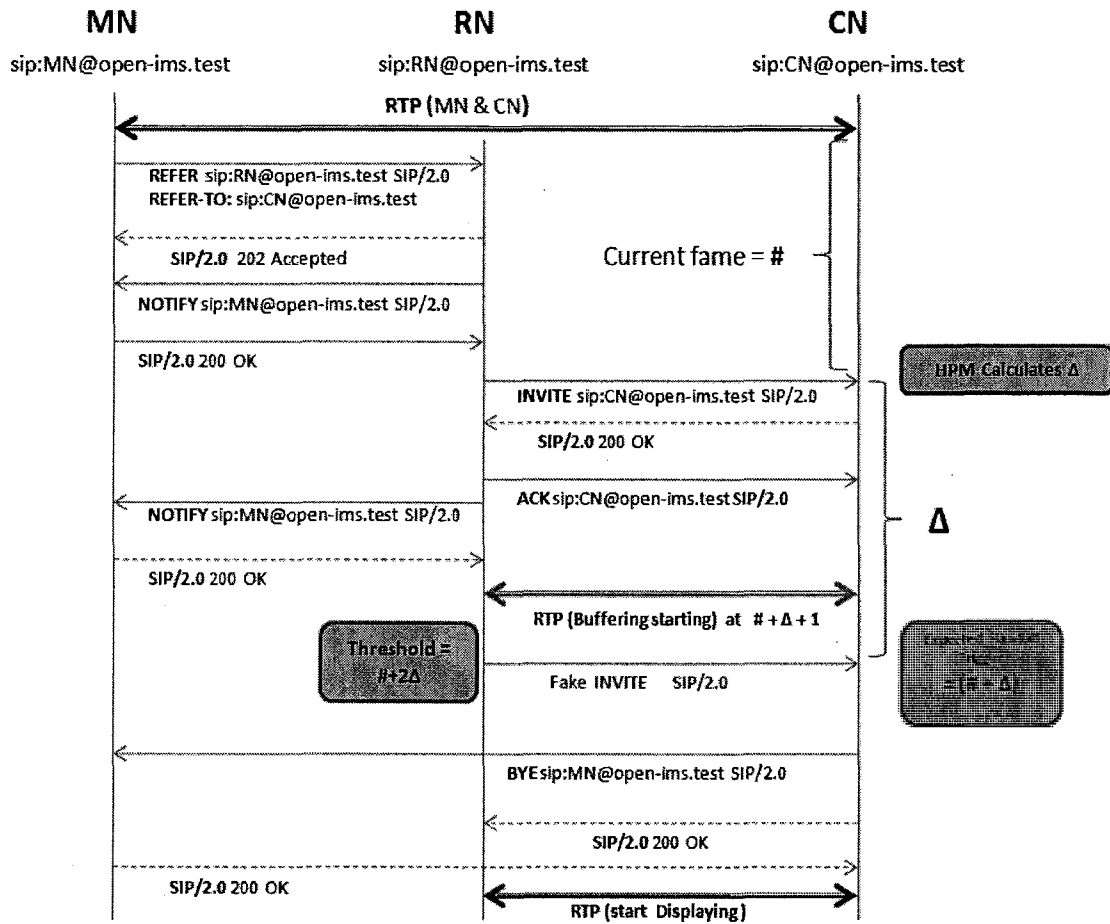


Figure 4.3 Call Flow of Session Transfer

HPM keeps track of the handoff delay history of the user saved, and updates it each time the user transfers a session. It retrieves streaming video parameters that include frame rate and frame size from the streaming server running on the CN to determine the number of video frames to be sent to the MN and RN during the handoff process.

CN keeps streaming video to MN starting at frame (#) normally, and simultaneously begins to buffer video to RN starting at frame ($\# + \Delta + 1$). The video will be buffered to RN until it reaches a specific threshold; the threshold is controlled by the Buffer

CHAPTER 4. ARCHITECTURE AND FEATURE

Controller running at RN, once the threshold equals $(\# + 2\Delta)$, RN sends a SIP FAKE INVITE to CN. Fake INVITE does not establish a real session, rather, it notifies CN, which is ready to start displaying the cached video at RN, and stop streaming video to MN. CN checks if the current frame sent to RN equals to $(\# + \Delta)$, and then sends a BYE message to MN to terminate the session; it simultaneously sends an OK message to RN to start displaying the cached video.

4.5.2 Mobile Node (MN)

The MN is the entity that starts the transfer of the video session. As shown in Figure 4.4, MN sends the REFER request to RN, and enters the “Transferring” state—Recall that the REFER-TO request contains the SIP URI of the CN.

When in the “Transferring” state, if all responses from corresponding referees RN are 202 Accepted, the MN transits to the “Proceeding” state. On the contrary, if RN does not have capabilities to process the REFER request or does not support the appropriate media type, it will reject the REFER request; or if the REFER request times out after specific period, the MN transits to the “Completed” state in order to stop the transfer mechanism.

According to [40], RN uses the NOTIFY request to report the progress of the session setup between RN and CN. MN stays in the “Proceeding” state as long as it receives provisional responses from RN. Upon receiving the final response 200 OK from RN, MN transits to the “Success” state, which signifies that RN has succeeded to establish the

CHAPTER 4. ARCHITECTURE AND FEATURE

video session to the CN. Otherwise, MN transits to state “Completed” to stop the transfer mechanism.

After the session has been established with RN, CN sends a BYE request to MN to terminate the old session and cause MN to enter the “Completed” state.

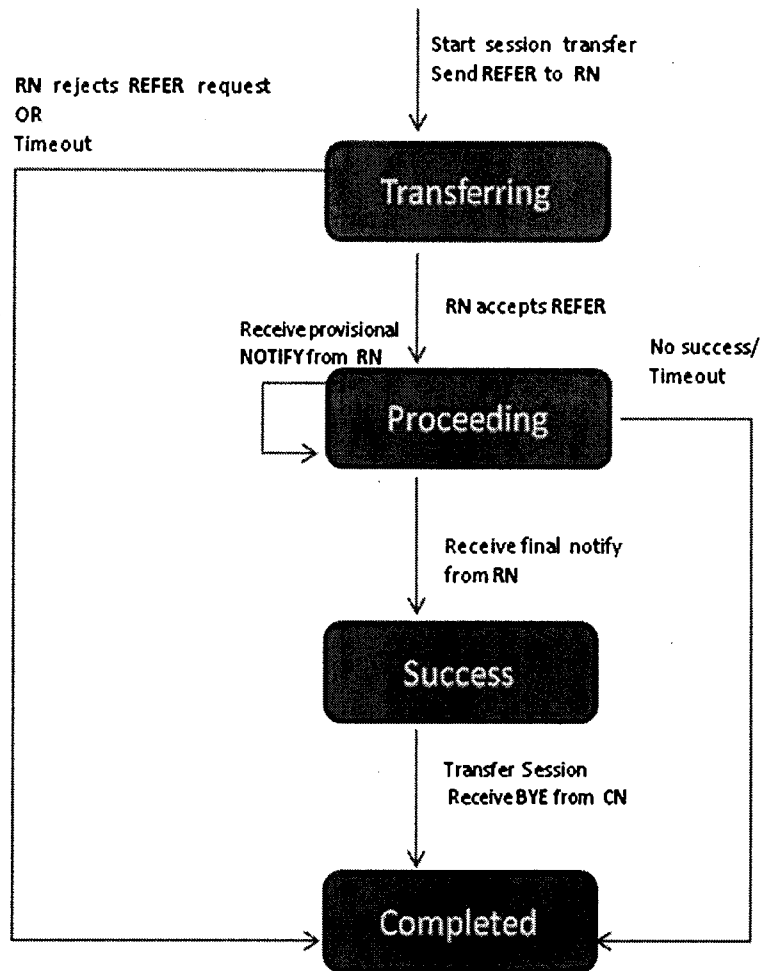


Figure 4.4 MN behaviour

4.5.3 Recipient Node (RN)

When RN receives the REFER request from MN, it enters the “Trying” state, as shown in Figure 4.5. RN will check the media parameter; if RN supports the media type assigned by REFER request, it sends 202 Accepted response to MN, and generates an appropriate SDP description for this type of media and put it into the body of the new INVITE request to be sent to CN—RN then enters the “Establishing” state. If RN does not support the media type assigned by REFER request, it will send a reject response to MN and transit state to “Completed”. If the INVITE request experiences timeout notification, RN will also transit to state “Completed”.

Upon receiving the buffering information from the CN, RN enters the “Buffering” state and keeps buffering until it reaches the specific threshold, at which point RN needs to notify CN that it is ready to start displaying the cached video and stop streaming video to MN; so RN sends FAKE INVITE to CN and enters the “Displaying” state. CN sends an OK response to RN to start displaying the video. Once the session is over, RN transits to the “Completed” state.

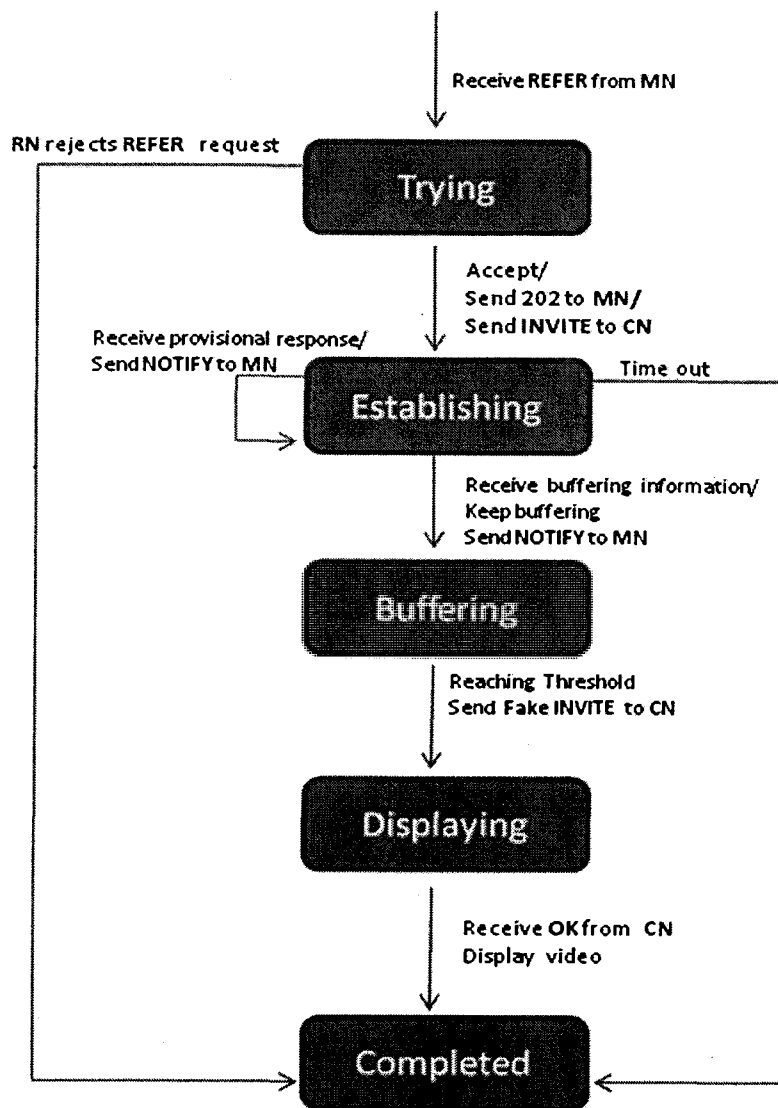


Figure 4.5 RN behaviour

4.5.4 Correspondent Node (CN)

When CN receives an INVITE request from RN, and if it supports the INVITE request, it enters the “Recording” state, as shown in Figure 4.5.; otherwise CN rejects the INVITE request and enters the “Completed” state.

Upon receiving the INVITE request from RN, CN calculates the predicted handoff time and sends the buffering information to RN, then ultimately transits to state “Streaming”.

Upon receiving the FAKE INVITE from the RN, CN sends an OK message to RN to start displaying the video, and a BYE message to MN to terminate the old video session and finally transits state to “Completed”.

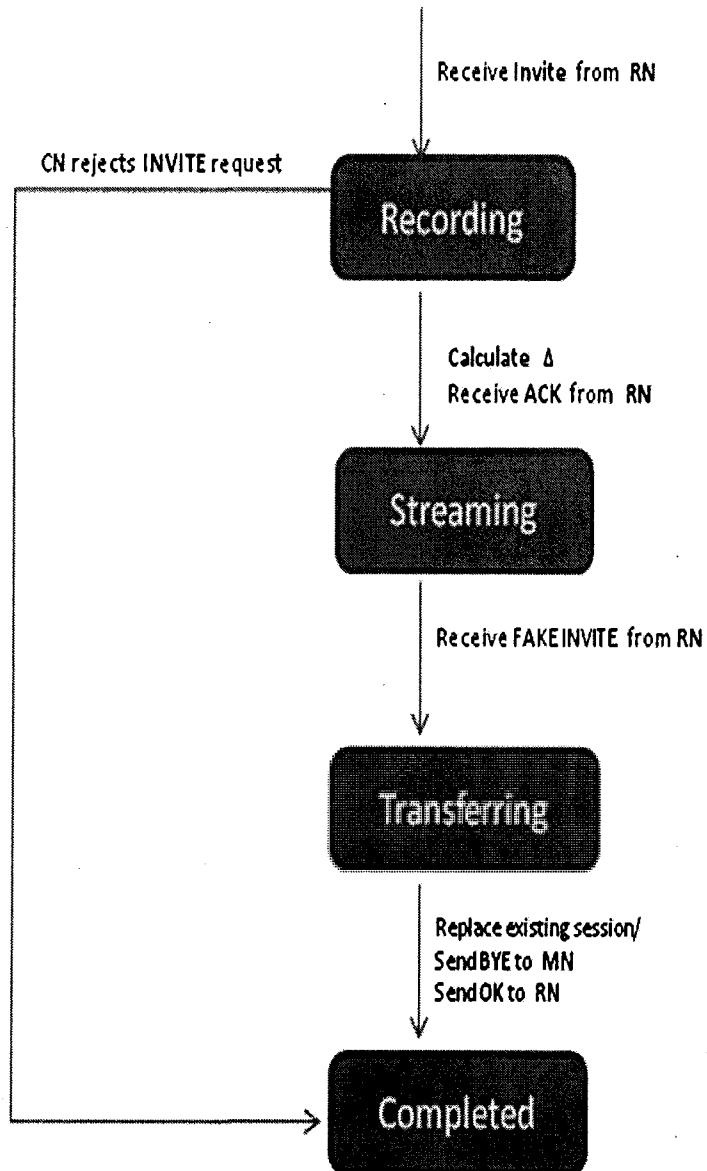


Figure 4.6 CN behaviour

4.6 Summary

In this chapter, we discussed the SIP-based handoff prediction manager and Buffer Controller in detail. We described the behavior and interaction between elements involved in the system; most importantly the Recipient Node, Mobile Node and Correspondent Node.

The next chapter describes the implemented system prototype. We discuss how the HPM and Buffer Controller are set up, and reveal tools used in this system and how the system is deployed. We also present the experimental environment, and describe the working prototype.

Chapter 5 System Implementation

This chapter presents a prototype implementation of our proposed Handoff Prediction manager; we introduce the tools and techniques employed in implementation, and provide snapshots and figures illustrating the prototype.

The system implementation is based on a three-stage process. The first stage involves the deployment and configuration of an IMS network; the second features the development of a SIP-based application to transfer the video session (which is compatible with IMS network); and the third and final stage encompasses the integration of video streaming capacities in this application.

5.1 Overview

There have been many SIP stacks, user agents, video streaming servers that have been implemented in the past years, and we decided to go with PJSIP framework, VLC media player, Gstreamer framework and Open IMS Core as the fundamental architecture in our HPM implementation, which we have built a test-bed in our research lab. All these tools are open source. A brief description of each tool will follow in the next section.

CHAPTER 5. SYSTEM IMPLEMENTATION

5.2 Tools

5.2.1 PJSIP Framework

PJSIP framework, [20], is an open source SIP stack and media stack for presence, instant messaging, and multimedia communication. Written in C, PJSIP has extensive documentation, high performance, cross-platform and portability compared to other SIP stacks. PJSIP components can be rapidly developed, tested and integrated into a variety of platforms with access to numerous tools and utilities. PJSIP API implements SIP Features (RFC 3261) and SIP extensions such as REFER request (RFC 3515) or Replaces header (RFC3891). PJSIP applications are lightweight, a very important aspect in seamless video transfer especially using mobile devices with poor CPU resources.

5.2.2 Open IMS Core

Open IMS Core, [19], is an implementation of IMS Call Session Control Functions (CSCFs) and a lightweight Home Subscriber Server (HSS). It is used as an architectural framework for delivering IP multimedia services over fixed and mobile networks using open standards to mobile users.

5.2.3 Gstreamer Framework

Gstreamer framework[18], GStreamer is a framework for creating streaming media applications. GStreamer pluggable components can be mixed and matched into arbitrary pipelines, thus making it possible to write a full-fledged video or audio editing application.

CHAPTER 5. SYSTEM IMPLEMENTATION

Gstreamer application has been developed and integrated with VLC mainly to control the stream video buffer at RN.

5.2.4 VLC Streaming Server

VLC streaming server[21] is a free cross-platform media player that supports a large number of multimedia formats, without the need for additional codecs. It can also be used as a streaming server, as shown in Figure 5.1. Extended features (video on demand, on the fly transcoding) can be also used to stream in unicast or multicast in IPv4 or IPv6 on a high-bandwidth network.

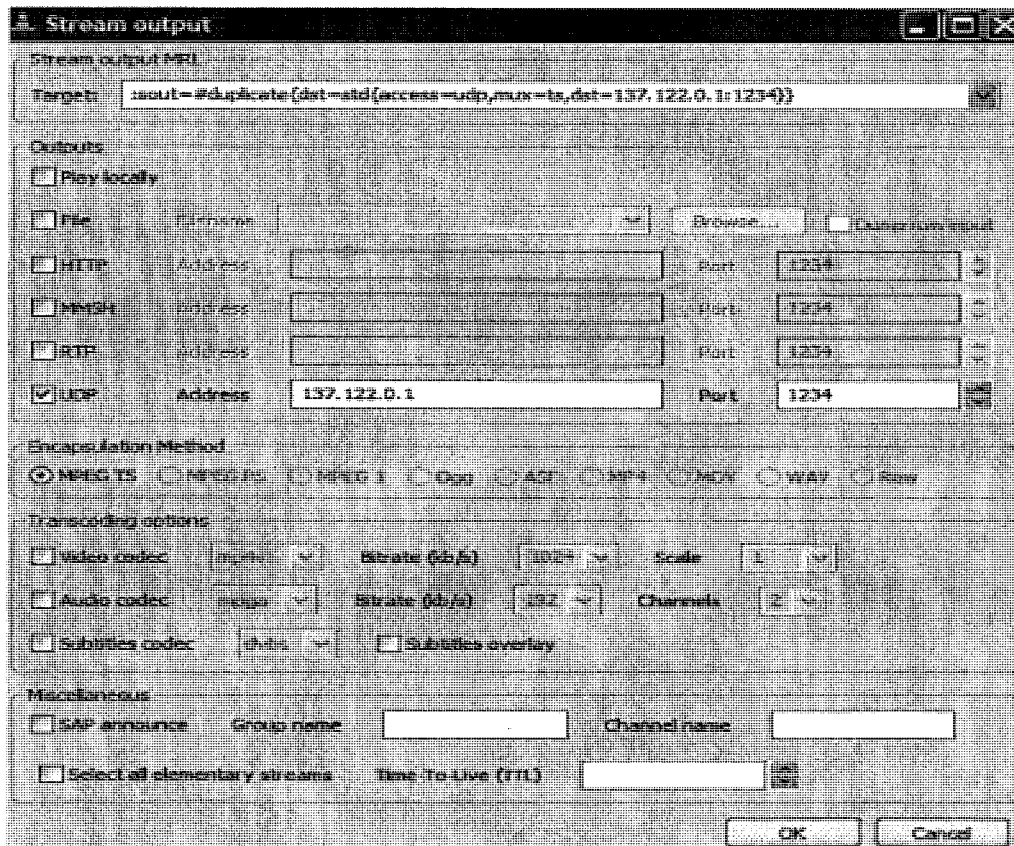


Figure 5.1 VLC Main Window

5.3 System Deployment

5.3.1 Open IMS Core deployment

The IMS network has been deployed using Open IMS Core. It is an open-source implementation of IMS proxies and HSS user database. Open IMS was developed by the Fraunhofer FOKUS Institute for Open Communication System, and is based on other open-source projects, such as the SIP Express Router (SER) for the proxies, MySQL for the HSS database and Tomcat to remotely administrate this network.

The Open IMS network has been deployed on one machine only, which played the role of the different SIP proxies (PCSCF, SCSCF and ICSCF), the HSS database, the HTTP administration server and the DNS server. Installation of the Open IMS network involved the following tasks:

- Installing GNU/Linux distribution (Ubuntu), since Open IMS is only available on Linux.
- Installing IMS source code from the SVN version from April 14th, 2008.
- Installing and configuring MySQL server for the HSS user database.
- Installing and configuring BIND DNS server so that the names of all proxies can be resolved, even though they are installed on the same machine and have the same IP address.
- Installing and configuring the Apache Tomcat HTTP server with Java Servlet.

5.3.2 PJSIP deployment

The purpose of using a PJSIP framework basically boils down to building a simple SIP User Agent on top of the PJSIP framework to allow us to register to the IMS network, establish a video session with VLC streaming server and finally transfer the video session seamlessly from the laptop to the desktop, as we will see in our testbed experiment.

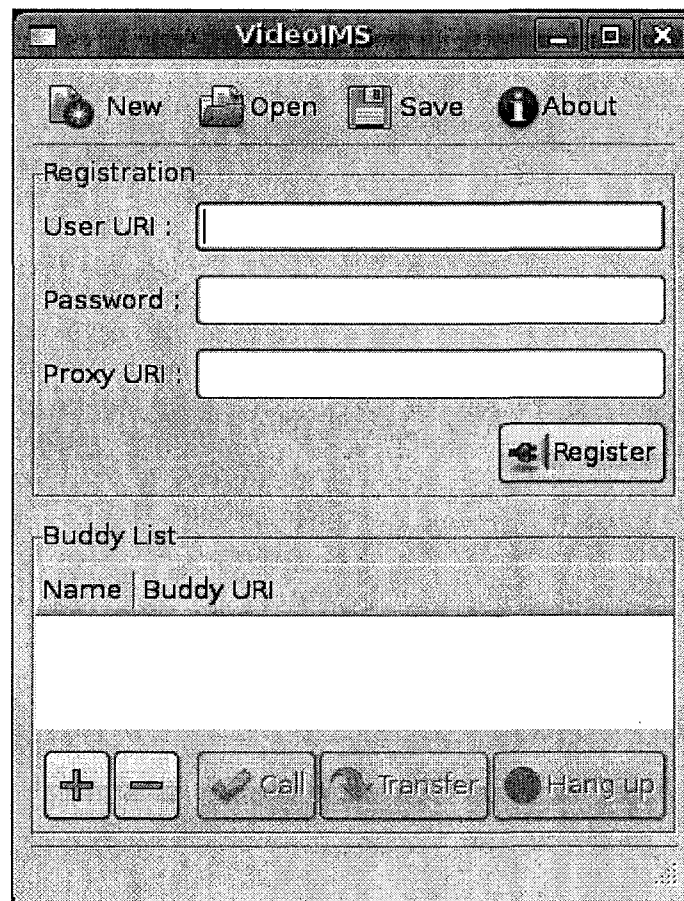


Figure 5.2 VideoIMS main screen

Figure 5.2 shows our implemented SIP UA (VideoIMS) [8], which supports most of the SIP methods INVITE, ACK, BYE, CANCEL, REGISTER, REFER and others.

CHAPTER 5. SYSTEM IMPLEMENTATION

Our SIP UA (VideoIMS) possesses all the functionality listed below; we created these functions using existing PJSIP framework API's.

- *init_sip_ua* (): initializes all the modules and memory pools needed for creating the UA.
- *worker_thread* (): dedicated for receiving SIP events/messages (outside or inside an existing dialog). Also passes the message to the appropriate function/callback.
- *register_acc* (): registers to the IMS network.
- *make_call* (): initializes a SIP call (sends an INVITE message to the video server to start a video stream).
- *transfer_call* (): transfers a SIP call.
- *refer_outside_dialog* (): transfers a SIP call using the SH method (sends a REFER message outside an existing dialog).
- *hangup_call* (): sends the BYE message to end a SIP call.
- *create_sdp_session* (): creates the SDP body in a SIP message (INVITE request or OK answer in general).
- *on_rx_request* (): called to handle incoming requests outside dialogs (e.g. INVITE message with or without REPLACES header).
- *on_call_transferred* (): handles incoming REQUEST messages.
- *call_on_state_changed* (): notifies changes in the INVITE session (call disconnected for example).
- *call_on_media_update* (): starts the media stream using VLC.

CHAPTER 5. SYSTEM IMPLEMENTATION

PJSUA is built over a high-level API (pjsua-lib) that is completely dependent on the media and codec API (PJMEDIA); therefore the existing PJSIP framework does not handle video streaming, and so implements its own Codecs. There are three approaches to resolve this problem:

1. Reusing PJMEDIA's Session Info.
2. Integrating Third Party Media Stack into PJMEDIA.
3. Completely Replace PJMEDIA with Third Party Media Stack.

Our implemented SIP UA (VideoIMS) replaces PJMEDIA with Third Party Media Stack (VLC), and integrates video streaming capabilities into PJSIP. This approach is supported by PJSIP developers since it does not change or modify the internal library code. It simplifies the structure of our VideoIMS client.

5.4 Experimental Environment

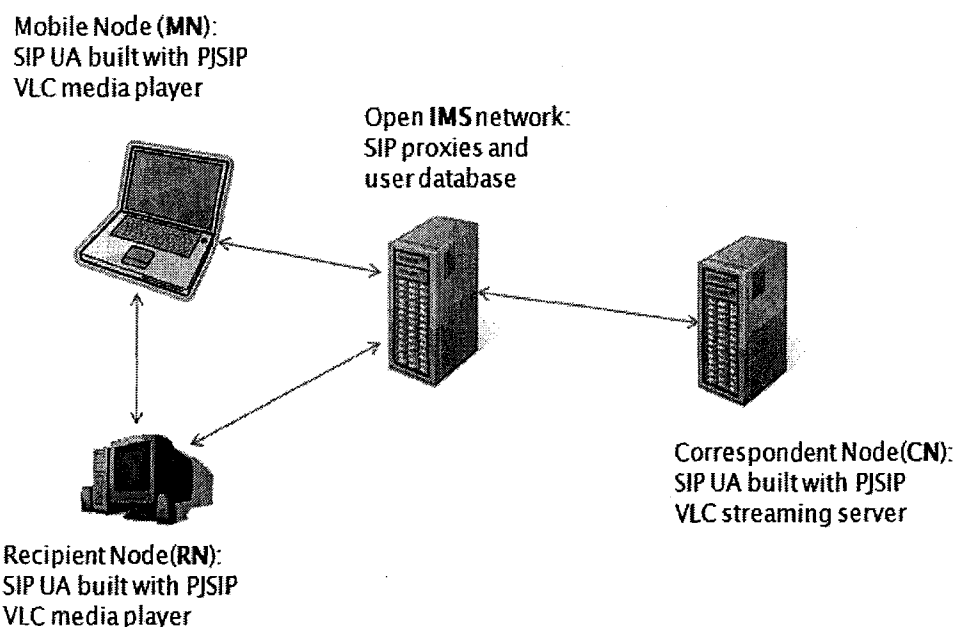


Figure 5.3 Experimental Environment

CHAPTER 5. SYSTEM IMPLEMENTATION

Our experimental environment, as shown in Figure 5.3, has the Open IMS core deployed in one Desktop computer that played the role of the different SIP proxies (PCSCF, SCSCF and ICSCF), the HSS database, the HTTP administration server and the DNS server, and that has been preconfigured to work with the domain “*open-ims.test*”. We prepared two more Desktop computers and a laptop to play the roles of **CN**, **RN**, and **MN** respectively.

- **MN** is registered in the IMS network as *alice@open-ims.test* with Public URI as *sip: alice@open-ims.test*.
- **RN** is registered in the IMS network as *bob@open-ims.test* with Public URI as *sip: bob@open-ims.test*.
- **CN** is registered in the IMS network as *video@open-ims.test* with Public URI as *sip: video@open-ims.test*.

VLC streaming server has been installed and configured on the **CN** only, whereas VLC media player has been installed on **MN** and **RN**.

5.5 Working Prototype Description

VideoIMS is a simple Graphical User Interface (GUI) built on top of our SIP UA. It has been developed by my co-worker Damien Herraud using GTK+, a highly usable feature-rich toolkit for creating graphical user interfaces. GTK+ boasts cross platform compatibility and an easy-to-use API. It is written in C, but has bindings to many other

CHAPTER 5. SYSTEM IMPLEMENTATION

popular programming languages, such as C#, Java, C++, and Python. It is free open source library.

As shown in Figure 5.2 [8], VideoIMS comprises of the following:

- **Buttons:** allow the user to register to the IMS network, add/remove buddies, establish a call, transfer a call or hang up a call.
- **Toolbar:** allows the user to open, save the configuration or create a new one, or to display the about dialog window.
- **Textboxes:** allow the user to enter the registration information.
- **Status bar:** informs the user of the status of his action.

The buttons are activated or deactivated according to what the user is authorized to do.

For examples, it is forbidden to:

- Call a buddy without being registered to the IMS network.
- Remove the buddy the user is currently calling.
- Transfer a call whereas no previous call is established.
- Hang up without any existing call.

5.5.1 Registering to IMS

Initially MN (Alice), RN (Bob), CN (video) launch their SIP UA (VideoIMS); enter the registration information (SIP URI, password and Proxy URI); and push the Register button to register to the IMS network. SIP UA validates the syntax, registers the users and displays the word registering at the status bar as shown in Figure 5.4.

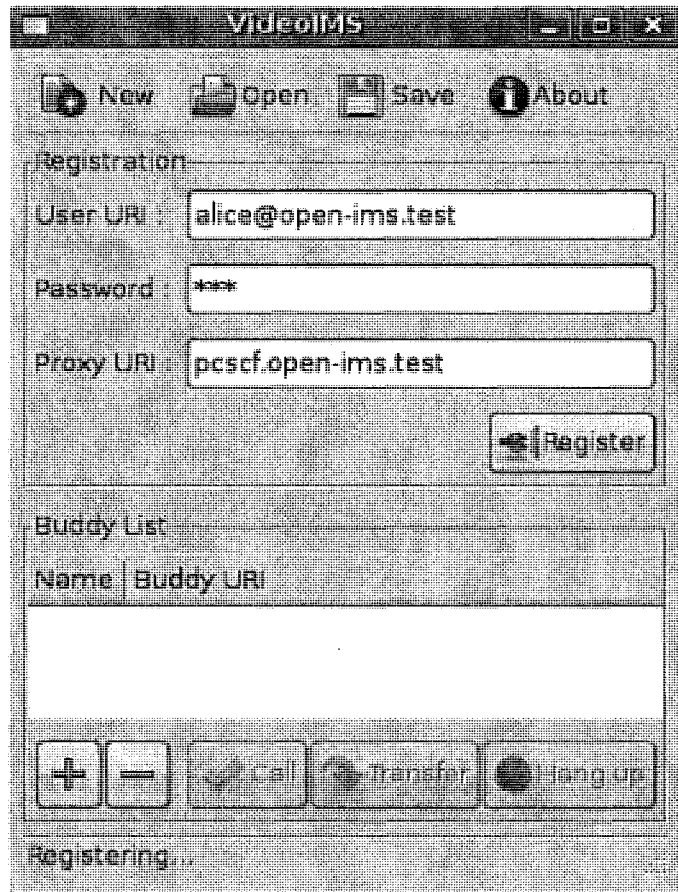


Figure 5.4 Registering to IMS network

Each user can push the Save tab to save the configuration information as text file. In the future, a user can open the file and register directly without entering any information. For example if Alice pushes the save button to save her registration information the text file will contain the information below:

```

User URI: alice@open-ims.test
Password: ****
Proxy URI: pcscf.open-ims.test
Buddy 1: bob::bob@open-ims.test
Buddy 2: video::video@open-ims.test

```

CHAPTER 5. SYSTEM IMPLEMENTATION

5.5.2 Adding buddy list

The second step, shown in Figure 5.5, sees the MN user adding CN and RN to the buddy list by clicking on the + button displaying buddy added on the status bar. MN can remove the items added to the buddy list by clicking on the – button.

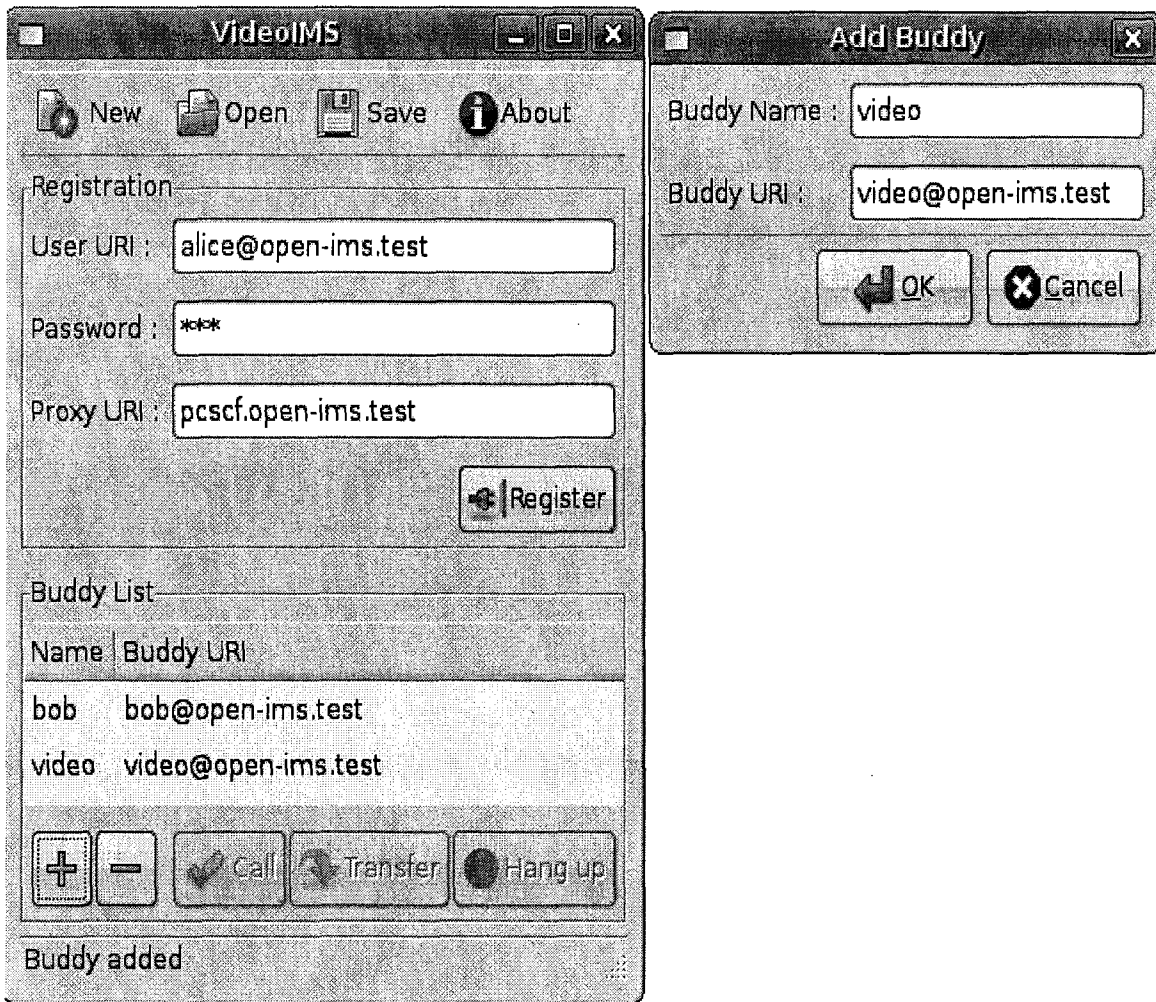


Figure 5.5 Adding CN, RN to buddy list

CHAPTER 5. SYSTEM IMPLEMENTATION

5.5.3 Establishing video session

In step 3, as shown in Figure 5.6, MN (Alice) selects CN (Bob) from the buddy list and push the Call button to establish the session and start streaming video.

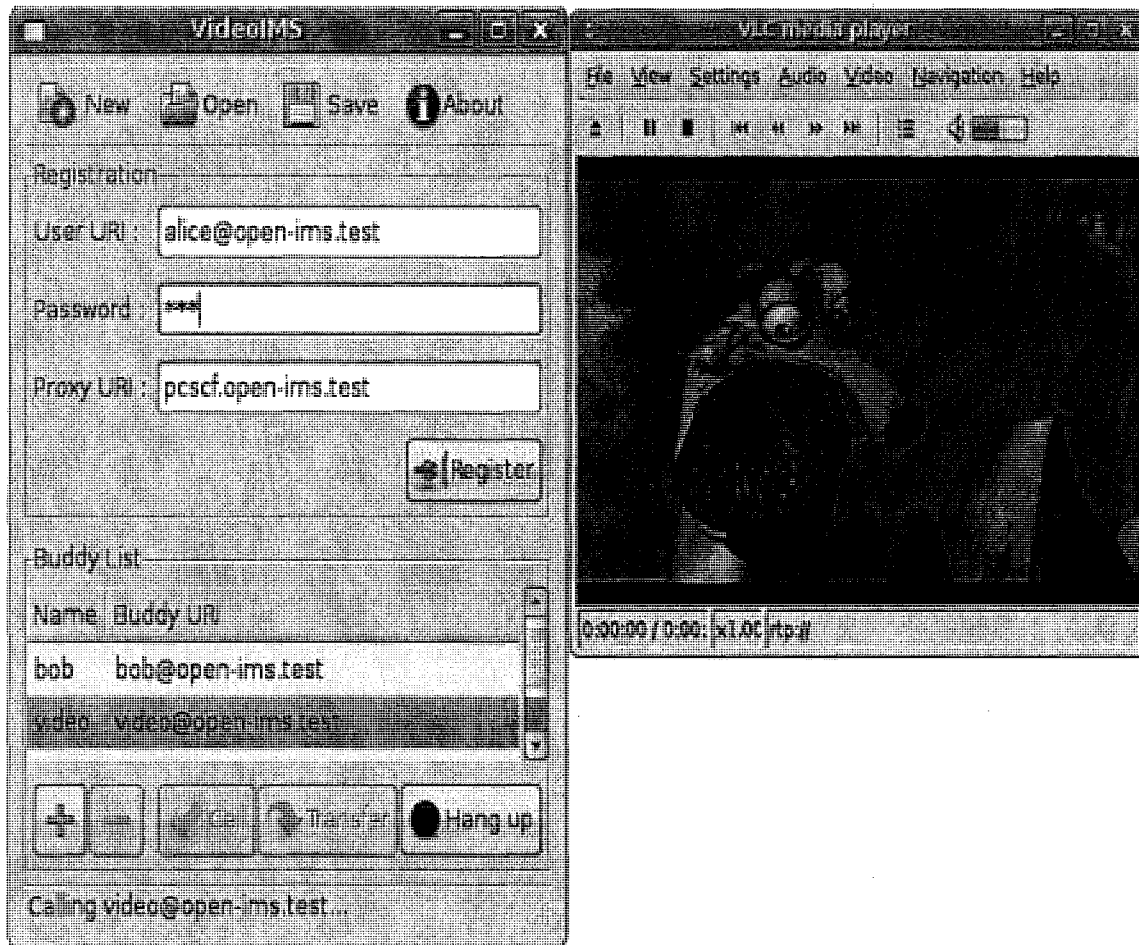


Figure 5.6 Establishing video session

5.5.4 Transferring video session

In the last step, shown in Figure 5.7, MN (Alice) selects RN (Bob) from the buddy list and pushes the Transfer button to establish the session transfer. This will cause the video

CHAPTER 5. SYSTEM IMPLEMENTATION

session to stop on the MN and resume on the RN. Now the video session has been seamlessly transferred from the MN to RN.

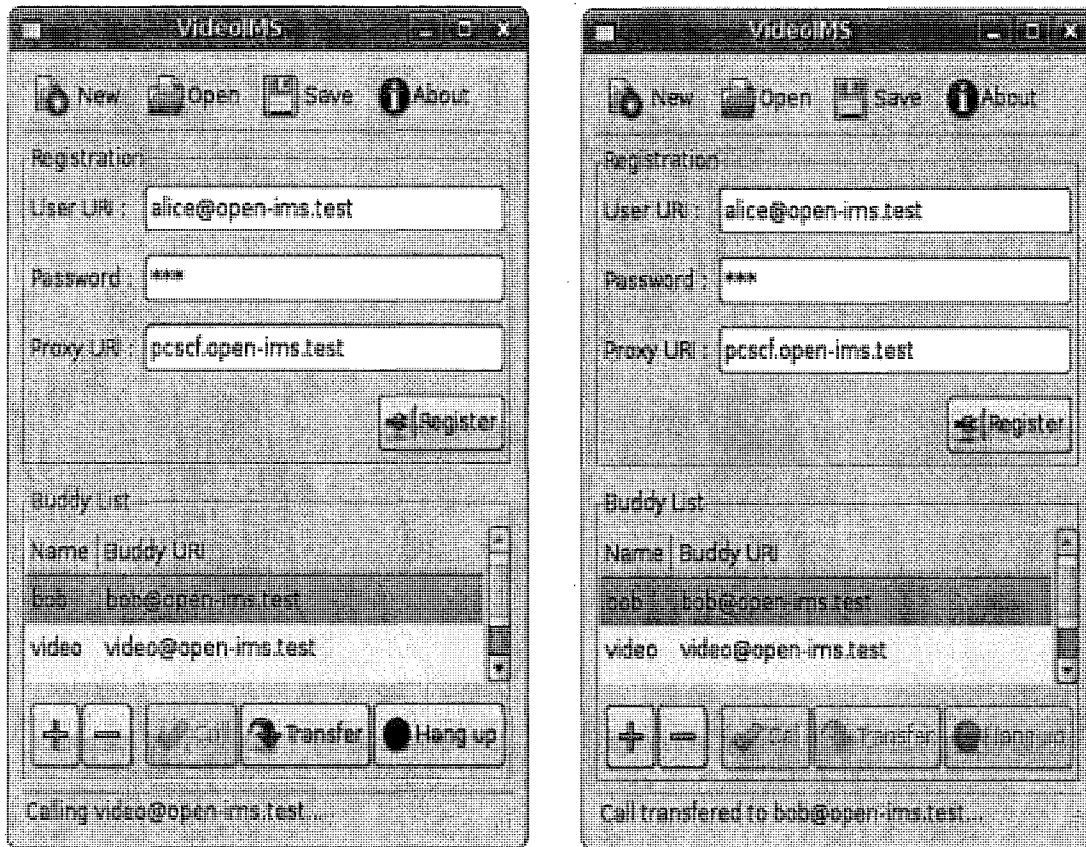


Figure 5.7 Transferring the video session

5.6 Performance Evaluation

The goal of our testbed was to transfer the video in session mobility over IMS network seamlessly without any disruption by predicting the handoff time. Our testbed application is very lightweight, requiring about 19 MB of RAM (4 MB for the PJSIP application plus 15 MB of RAM for VLC to stream or play the video). Session transfer has been tested in a LAN network. The total handoff time was about 925 ms, which included 25 ms for the SIP message flow to transfer the session, plus 900 ms to terminate the VLC video stream on the MN (the transferor of the session), and resume the VLC video stream on the RN (the transferee) on Intel Pentium IV 3.4 GHz. This duration depends on the CPU power of the machine.

The graph in Figure 5.8 shows that the predicted values are high at the beginning since the user's handoff history contains few values only; the more values saved to the user's history, the more accurate the prediction is.

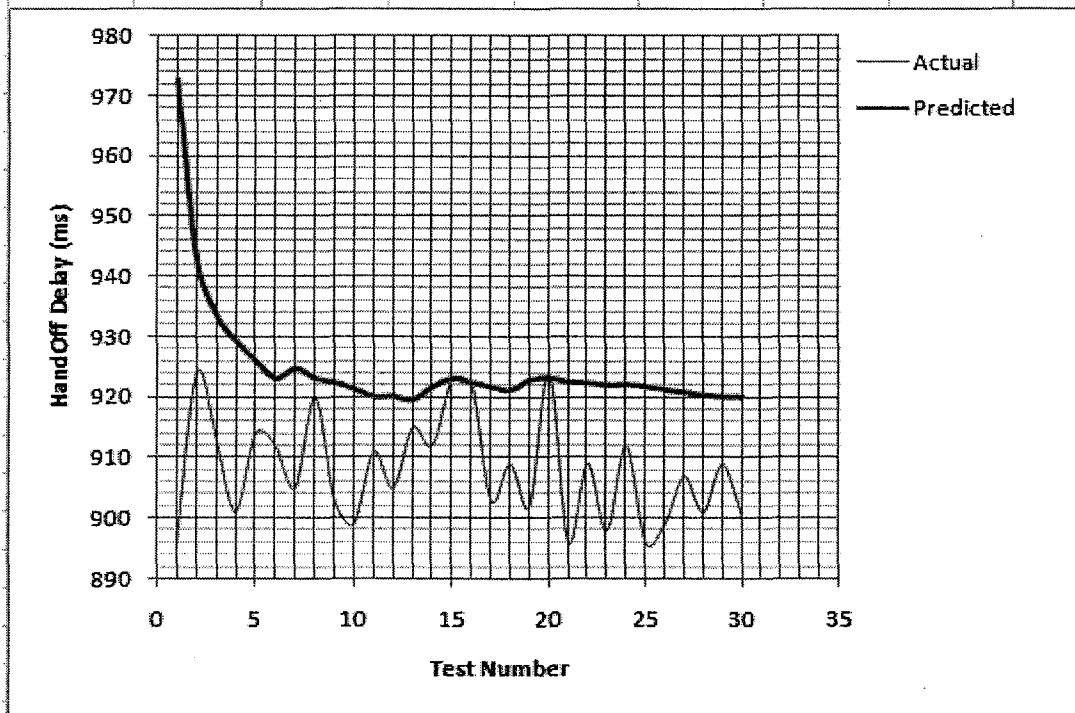


Figure 5.8 Predicted Handoff Time vs. Actual

As shown in Figure 5.9, our predicted handoff values *at* $k = 0.16$ were very close to the actual handoff values, which guarantees no frame loss during the handoff process the difference between the values is almost zero.

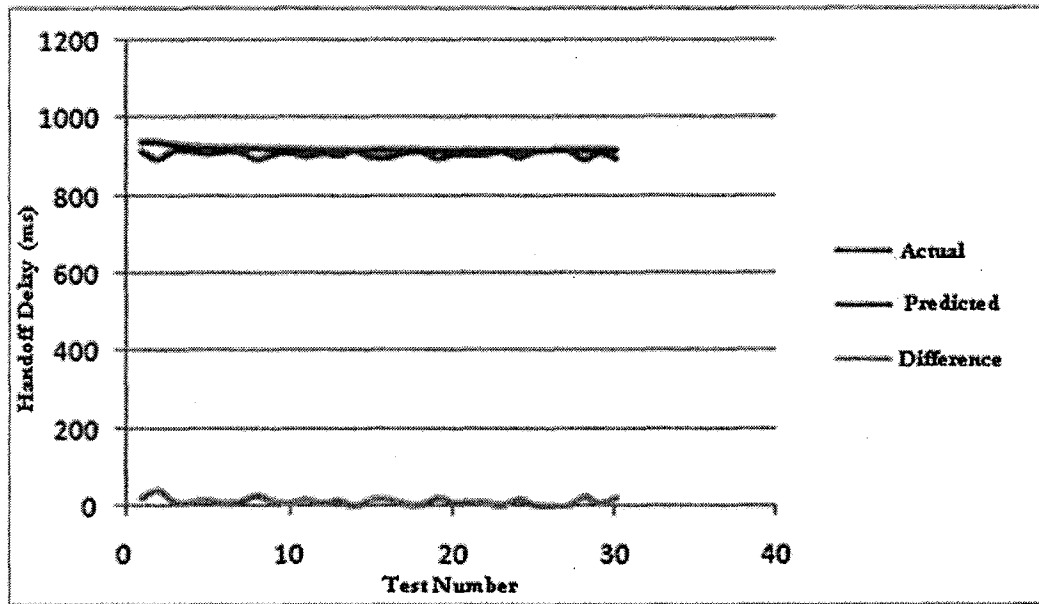


Figure 5.9 Differences between Predicted and Actual Handoff Time

5.7 Summary

We presented our prototype implementation for Handoff Prediction Management system. In that effort, we described the system prototype implementation details and the different tools and techniques utilized in the implantation were elaborated. We included snapshots and graphs that showed the results obtained, and which proved that we have made good decisions in choosing the platform and tools to implement the system.

Chapter 6 Conclusion and Future Work

6.1 Conclusion

Mobile devices have been gaining on popularity, and the last few years have shown a noticeable improvement of handheld devices capacities, enabling them to support large multimedia services, applications and IP-based wireless and wired connectivity. Ultimately, users of these devices will be able to access IP multimedia content on the internet because they will easily move within a wireless domain and enjoy multimedia services. For the time being however, these wireless handheld devices are still limited in term of autonomy, display capacity, ease of use and computational power, and stationary devices, like PC continue to be more adapted for multimedia services.

Providing seamless handoff in session mobility is an important and challenging issue in the heterogeneous next generation All-IP networks. Most of the existing approaches and topologies for session mobility suffer from a major problem in transparency, making the user experience a lapse in communication during handoff time. Some solutions cause a segment of media to be redundant (from the time that the new media session was set up successfully until the old media session was terminated).

CHAPTER 6. CONCLUSION AND FUTURE WORK

To accommodate the growing demand in session mobility, and to address deficiencies in existing architectures, our thesis proposes a new framework to achieve seamless video handoff in session mobility over the IP Multimedia Subsystem (IMS) network. The framework is built on top of the IMS network. IMS seems to be the best platform for session mobility deployment because it merges mobility and internet services such as video streaming. The framework uses SIP REFER method to transfer the SIP session, and HPM to determine the number of video frames to be streamed to the MN and RN during the handoff time. This feat is achieved by predicting the required handoff time while transferring the video session from the MN to the RN. The REFER method, combined with HPM, allows the mobile user to transfer the video session seamlessly without any disruption.

During the handoff process, the user watches the video at the MN. Simultaneously, CN buffers the video to RN, rendering the handoff process seamless and transparent. HPM keeps track of the handoff delay history of the user, and updates it each time the user transfers a session.

We have also implemented a SIP UA (VideoIMS) with the ability to transfer a video session; the implementation results show effective seamless handoff in session mobility without any disruption.

6.2 Future work and suggestions

The door is wide open for future improvements; firstly, we would try to enhance the functionality of HPM to accommodate situations where predicted handoff time is underestimated. However, experiments that were performed in our lab so far have showed that the predicted handoff times calculated by the existing HPM were always equal to or greater than the actual handoff time

Another improvement would be to implement a discovery mechanism where the mobile user is able to locate the devices that are available in his locality.

Enhancing our SIP UA to implement automatic transfer by proximity detection is yet another sought-after enhancement. This may be achieved through RFID tags attached to the MN. No later than the RFID reader detects that the MN is within the RN range, it would signify a trigger point for the MN to start transferring the session to the RN

Still in the realms of SIP, we plan to deploy a SIP registrar that stores the capabilities of each device. For example, a device that has projection capabilities could register itself as a projection device; later when a mobile user looks for an appropriate device, the SIP registrar can be queried to identify such a device.

Also pertaining to SIP UA, we believe it can be enhanced to have the ability to split the session into multiple devices, and to retrieve the session back if required

CHAPTER 6. CONCLUSION AND FUTURE WORK

We plan to test our SIP UA on a WiMAX network and compare the evaluation results to existing counterparts, as our experiments were tested on a LAN network.

And finally, we also plan to enhance our SIP UA to consider security issues such as malicious attacks by encrypting some headers fields within the SIP requests as proposed in [23].

Bibliography

- [1] 3GPP, "Network architecture," Release 6, TS 23.002, Jun. 2005.
- [2] A. McDonald, and T. Znati, "A Mobility-based Framework Adaptive Clustering in Wireless Ad-Hoc networks," *IEEE Journal on selected Areas in Communications*, vol. 17, no 8, Aug. 1999.
- [3] A. Dutta, S. Madhani, W. Chen, O. Altintas, and H. Schulzrinne, "Optimized Fast-Handoff Schemes for Application Layer Mobility Management," *Mobile Computing and Communications Review (MC2R)*, vol. 7, no. 1, pp. 17-19, Jan. 2003.
- [4] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging," RFC 3428, Internet Engineering Task Force, Dec. 2002.
- [5] B. Park, et. al., "Mobility Management using SIP in All-IP networks," Wireless World Research Forum (WWRF), KOREA.
- [6] C. Perkins, "IP Mobility Support," RFC 3344, Internet Engineering Task Force, Aug. 2002.
- [7] C. Politis, K. Chew, and R. Tafazolli, "Multilayer Mobility Management for All-IP Networks: Pure SIP vs. Hybrid SIP/Mobile IP," *IEEE VTC2003 spring*, vol. 4, pp. 2500-2504, Apr. 2003.
- [8] D. Herraud, A. Karmouch "Design and implantation of SIP-based mobility protocol," Technical Report, IMAGINE lab at University of Ottawa, Aug. 2008.

- [9] D. Marples, "Naming and accessing internet appliances using extensions to the session initiation protocol," in Proc. of SIP 2000 Conference and Exhibition, France, May 2000.
- [10] D. Vali, S. Paskalis, A. Kaloxylos, and L. Merakos, "An Efficient Micro-Mobility Solution for SIP Networks," *GLOBECOM2003*, vol. 22, no. 1, pp. 3088-3092, Dec. 2003.
- [11] E. Wedlund and H. Schulzrinne, "Mobility support using SIP," in Proc. of the 2nd ACM international Workshop on Wireless Mobile Multimedia. WOWMOM '99, New York, Aug. 1999.
- [12] F. Vakil, A. Dutta, J-C. Chen, M. Tauil, S. Baba, N. Nakajima, Y. Shobatake, and H. Schulzrinne, "Supporting Service Mobility with SIP," Internet Draft, Internet Engineering Task Force, Dec. 2000.
- [13] H. Schulzrinne, and E. Wedlund, "Application-Layer Mobility using SIP," *Mobile Computing and Communications Review (MC2R)*, vol. 4, no. 3, pp. 47-57, Jul. 2000.
- [14] H. Schulzrinne, and J. Rosenberg, "Internet telephony: Architecture and protocols – an IETF perspective," *Computer Networks and ISDN Systems*, vol. 31, pp. 237–255, Feb. 1999.
- [15] H. Schulzrinne and J. Rosenberg, "The session initiation protocol: Internet-centric signaling," *Communications Magazine, IEEE*, vol.38, no.10, pp.134-141, Oct. 2000.
- [16] H. Schulzrinne S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, Internet Engineering Task Force, Jan. 1996.
- [17] H. Schulzrinne, "Personal mobility for multimedia services in the Internet," European Workshop on Interactive Distributed Multimedia Systems and Services (IDMS), Berlin, Germany, Mar. 1996.

- [18] <http://gststreamer.freedesktop.org/>
- [19] <http://www.openimscore.org/>
- [20] <http://www.pjsip.org/>
- [21] <http://www.videolan.org/>
- [22] I. Popescu, "Supporting Multimedia Session Mobility using SIP," Communication Networks and Services Research conference CNSR 3003, New Brunswick, Canada 2003.
- [23] J. Peterson, "Session Initiation Protocol (SIP) Authenticated Identity Body (AIB) Format," RFC 3893, Internet Engineering Task Force, Sept. 2004.
- [24] J. Rosenberg, "A Presence Event Package for the Session Initiation Protocol (SIP)," RFC 3856, Internet Engineering Task Force, Aug. 2004.
- [25] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, Internet Engineering Task Force, Jun. 2002.
- [26] J. Rosenberg, J. Peterson, H. Schulzrinne, and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)," RFC 3725, Internet Engineering Task Force, Apr. 2004.
- [27] K. Wong, A. Dutta, J. Burns, R. Jain, K. Young, and H. Schulzrinne, "A Multilayered Mobility Management Scheme for Auto-Configured Wireless IP Networks," *IEEE Wireless Communications*, vol. 10, no. 5, pp. 62-69, Oct. 2003.
- [28] K. Ohta, T. Yoshikawa, T. Nakagawa, and S. Kurakake, "Adaptive Terminal Middleware for Seamless Session Mobility," *IEICE Transactions on Information & Systems*, vol. E86-D, no. 11, pp. 2343-2352, Nov. 2003.

- [29] L. Rong, and I. Burnett, "Application Level Session Handoff Management in Ubiquitous Multimedia Environment," *e-Business and Telecommunication Networks*, pp 298-304. Aug. 2006.
- [30] M. Chen, and F. Wang, "Session mobility of SIP over multiple devices," in Proc. of the 4th international Conference on Testbeds and Research infrastructures For the Development of Networks & Communities (ICST), Belgium, Mar. 2008.
- [31] M. Chen, C. Peng, and R. Hwang, "SSIP: Split a SIP Session over Multiple Devices," *Computer Standards and Interfaces*, vol. 29, no. 5, pp 531-54, Jan. 2007.
- [32] M. Handley, and V. Jacobson, "SDP: Session Description Protocol," RFC 2327, Internet Engineering Task Force, Apr. 1998.
- [33] M. Bargh, H. Zandbelt, and A. Peddemors, "Managing Mobility in Beyond-3G Environments," *Lecture Notes in Computer Science (LNCS)*, vol. 3079, pp. 545-555, 2004.
- [34] N. Akhtar, M. Georgiades, C. Politis, and R. Tafazolli, "SIP-based end system mobility solution for all-IP infrastructures," in Proc. of the IST mobile & wireless communications summit, 2003.
- [35] N. Freed and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," RFC 2045, Internet Engineering Task Force, Nov. 1996.
- [36] P. Calhoun, J. Loughney, E. Guttman, and J. Arkko, "Diameter Base Protocol," RFC 3588, Internet Engineering Task Force, Sept. 2003.
- [37] P. Faltstrom, "E.164 number and DNS," RFC 2916, Internet Engineering Task Force, Sept. 2000.

- [38] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, Internet Engineering Task Force, Jun. 1999.
- [39] R. Shacham, H. Schulzrinne, S. Thakolsri, and W. Kellerer, "Session Initiation Protocol (SIP) Session Mobility," IETF Draft, version 5, Nov. 2007.
- [40] R. Sparks, "The Session Initiation Protocol (SIP) Refer Method," RFC 3515, Internet Engineering Task Force, April 2003.
- [41] S. Moyer, D. Maples, and S. Tsang, "A Protocol for Wide-Area Secure Network Appliance Communication," *Communications Magazine, IEEE*, vol.39, no.10, pp.52-59, Oct. 2001.
- [42] S. Moyer, D. Maples, S. Tsang, and A. Ghosh, "Service Portability of Networked Appliances," *Communications Magazine, IEEE*, vol.40, no.1, pp.116-121, Jan. 2002.
- [43] S. Moyer, D. Marples, S. Tsang, J. Katz, P. Gurung, T. Cheng, A. Dutta, and H. Schulzrinne, "SIP for Light Bulbs Using SIP to Support Communication with Networked Appliances," Internet Draft, Internet Engineering Task Force, Jul. 2000.
- [44] T. Choi and k. Chung, "A Session Mobility Scheme for Seamless VOD Service in Ubiquitous Environments," *Embedded and Ubiquitous (EUC)*, vol. 3207/2007, pp. 652-661, Jul 2004.
- [45] W. Elleuch, A. Houle, and S. Guenette, "Enabling Session Mobility in Full Mesh Conferencing Model," *Wireless and Mobile Computing, Networking and Communication (WiMOB). Third IEEE International Conference on*, vol., no., pp.49-49, 8-10 Oct. 2007.
- [46] Y. Cui, K. Nahrsetedt, and D. Xu, "Seamless user-level handoff in ubiquitous multimedia service delivery," *Multimedia Tools and Applications Journal*, vol. 22, no. 2, pp. 137-170, Feb. 2004.

- [47] Y. Zhang, "SIP-based VoIP network and its interworking with the PSTN," *Electronics & Communication Engineering Journal*, vol.14, no.6, pp. 273-282, Dec. 2002.