UNIVERSITY OF OSLO
UniK - University Graduate Center

Session Continuity in Heterogeneous Networks: A SIP-based Proactive Handover Scheme

Master thesis

Håkon Eyde Kjuus

August 1, 2007

i Abstract

Today, the computation power and storage capability on mobile devices are increasing rapidly, and together with new interactive services this creates a demand for more bandwidth. To keep track of this evolution, the use of heterogeneous networks has gained focus. Instead of only using one type of network, it is desired to utilize all carriers available on a device, and hence choose the one best suited.

This thesis describes and discusses different approaches to session continuity. The different solutions include Mobile IP, Generic Access Networks (GAN), and proposals based on the Session Initiation Protocol (SIP). Current solutions are not satisfactory in the terms of handover delay, and hence best suited for non-realtime applications.

Furthermore it is proposed an application-layer handover scheme for session continuity in heterogeneous networks, which will significantly reduce the handover time. This handover scheme has been implemented, and tests show that the handover time is significantly reduced.

Keywords: Session Continuity, Handover, Heterogeneous Networks, SIP, VoIP, GAN.

ii Sammendrag

Prosseseringshastigheten og lagringskapasiteten på mobile enheter øker hurtig, og sammen med nye interaktive tjenester skaper dette et behov for mer båndbredde. For å holde følge med denne utviklingen, har bruk av hetrogene nett fått øket oppmerksomhet. Istedet for kun å benytte en type nettverk, er det ønsket å kunne utnytte alle nettverk som er tilgjengelig på en terminal, og velge det som er best egnet til enhver tid.

Denne avhandlingen beskriver og diskuterer ulike tilnærminger til sesjonskontinuitet. De forskjellige løsningene inkluderer Mobil IP, Generic Access Networks (GAN), og forslag basert på Session Initiation Protocol (SIP). Eksisterende løsningen er ikke tilfredstillende når det kommer til handover forsinkelse, og er dermed dårlig egnet til til sanntids kommunikasjon.

Videre foreslås det et applikasjonslags handover-system for sesjonskontinuitet i hetrogene nett, som signifikant reduserer handovertiden. Denne løsningen er implementert, og tester viser at handover tiden blir signifikant redusert.

Nøkkelord: Sesjonskontinuitet, Handover, Hetrogene nett, SIP, VoIP, GAN.

iii Foreword

This report is the fulfillment of the requirements for the degree Master of applied Informatics at the University of Oslo. The thesis was written at UniK - University Graduate Center at Kjeller, under the supervision of Professor Torleiv Maseng.

I would like to thank Torleiv Maseng for suggesting this area of research, and for providing contact with actors from the communication industry. The inputs to the discussions from these parties have been very useful. In that relation I want to thank Lars Bråten from FFI and the people from Telenor R&I; Paal Engelstad, Thomas Haslestad, Hans Erik Karsten and Anne Mari Nordvik.

I also want to thank Alan Duric and his collegues from Telio, and Paal-Erik Martinsen and his collegues from Tandberg, who has given great input to the solution design.

I would also like to thank Dinko Hadzic from FFI for valuable input on the implementation work.

Finally, a special thanks to Elin Sundby Boysen for great collaboration and for implementing the server-side of the proposed solution.

Kjeller, July 2007

Håkon Eyde Kjuus

iv List of abbreviations

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
ATM	Asynchronous Transfer Mode
B2BUA	Back to Back User Agent
BSC	Base Station Controller
CN	Correspondent Node
CS	Circuit Switched
FA	Foreign Agent
FER	Frame Error Rate
FMC	Fixed Mobile Convergence
FMC	Fixed Mobile Convergence
GAN	Generic Access Network
GANC	GAN Controller
GERAN	GSM EDGE Radio Access Network
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
GUI	Graphical User Interface
GW	GateWay
HA	Home Agent
HLR	Home Location Register
HO	Handover
HTTP	HyperText Transfer Protocol
IAX	Inter Asterisk eXchange
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IM	Instant Messaging
IP	Internet Protocol
ITSP	Internet Telephony Service Provider
ITU	International Telecommunications Union
LAN	Local Area Network
MIP	Mobile IP
MN	Mobile Node
NAP	Network Attachment Point

NAT	Network Address Translation
PBX	Private Branch eXchange
POTS	Plain Old Telephone Service
PS	Packet Switched
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RAN	Radio Access Network
RFC	Request For Comments
RTP	Real-time Transport Protocol
RTT	Round-Trip Time
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SRTP	Secure Real-time Transport Protocol
TCP	Transmission Control Protocol
TETRA	Terrestrial Trunked Radio
TLS	Transport Layer Security
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
UMA	Unlicensed Mobile Access
URI	Uniform Resource Identifier
UTRAN	UMTS Radio Access Network
VoIP	Voice over IP
WLAN	Wireless Local Area Network

Contents

i Abstract i
ii Sammendrag ii
iii Foreword iii
iv List of abbreviations iv

1 Introduction **1**
1.1 Heterogeneous Networks and Mobility 1
1.2 Scope 2
1.3 Methods 2
1.4 Outline of the thesis 3
1.5 Reading the thesis 3

2 Background **5**
2.1 Heterogeneous networks 5
2.2 Mobility 7
2.2.1 Handover 8
2.3 Voice over IP (VoIP) 9
2.3.1 Session Initiation Protocol (SIP) 10
2.3.2 Other VoIP signaling protocols 13

3 Approaches to Mobility **15**
3.1 Generic Access Network (GAN) 15
3.1.1 Overview 15
3.1.2 Technology 16
3.2 SIP-based solutions 18
3.2.1 Overview 18
3.2.2 SIP in conjunction with GSM 18
3.2.3 Other SIP Mobility proposals 20
3.3 Mobile IP (MIP) 22
3.4 Mobility Methods Comparison 24
3.4.1 Choice of mobility solution 25

4 SIP-based Proactive Handover **27**
4.1 Introduction 27
4.2 Solution overview 28
4.2.1 Handover Procedure 30
4.3 Technical solution 31
4.4 Presumptions and Limitations 33

5 Implementation **35**
5.1 Overview 35
5.2 Multiple registrations 37
5.3 Call setup 39
5.4 2-Instance Implementation 39
5.5 Handover Procedure 41

6 Results and Evaluation **43**

7 Conclusion **47**
7.1 Further research 48

Appendices **53**

A Source code **53**

B Measuring **55**

C SIP message format **59**
C.1 Format 59
C.2 Requests 59
C.3 Responses 60

D Wireshark traces **61**

List of Figures

2.1 Mobility vs. bandwidth 6
2.2 Typical VoIP usage 10
2.3 Standard SIP with proxy 11
2.4 Back to Back User Agent 12

3.1 GAN architecture 16
3.2 GAN functional architecture[1] 17
3.3 SIP in conjunction with GSM 19
3.4 Handover Wedlund [2] 21
3.5 Handover Banerjee [2] 22
3.6 Handover Bellavista [2] 23
3.7 Mobile IP 23
3.8 IP encapsulation in MIP 24

4.1 Solution overview 29
4.2 Handover scenario with gradual degradation [2] 31
4.3 Handover scenario with broken primary link [2] 32
4.4 Technical solution 32

5.1 2 SIP Stacks 36
5.2 Signalling for direct hold 38
5.3 Multiple Registrations 40
5.4 2 SIP-Com. instances 40

D.1 Wireshark trace 61

List of Tables

2.1 Different wireless systems[3][4] 7

B.1 Results, high CPU speed 56
B.2 Calculated values, high speed 56
B.3 Results, low CPU speed 57
B.4 Calculated values, low speed 57

Chapter 1

Introduction

Today, there is a significant trend toward using not only *one* wireless network, but to utilize all carriers available on a mobile device. In addition, for real-time applications like voice, it is desired to make a seamless change of network without user interaction.

1.1 Heterogeneous Networks and Mobility

Heterogeneous Networks means networks of different types, for instance WLAN and GSM. The arrival of new bandwidth consuming services and mobile devices with more computation power generates a demand for more bandwidth on mobile devices[5]. For these reasons, support for heterogeneous networks is an attractive solution due to the possibilities of always using the most adequate carrier available.

The challenge in this context will be to provide change of network without user interaction, and in particular for real-time applications like voice, make a seamless handover between the different networks.

The level of mobility is often divided into four categories as further described in section 2.2. Voice over IP (VoIP) would be placed in the category Session Mobility[6]. This implies that the user is able to access the service anywhere, with any terminal, using different Network Access Points (NAP), and even maintain an active session while switching between terminals.

However, when extending VoIP to mobile devices, the initial mobility support is not sufficient. Mobile users will demand a mobility support equivalent to that known from existing cellular systems or better, that is, seamless change of NAP. This principle is known as seamless handover[3]. In addition, it is attractive to support not only planned and controlled handovers, but be able to maintain a

session even during link-breakage using other carriers available.

1.2 Scope

In this thesis I intend to address the issue of maintaining a VoIP session while changing Network Access Point (NAP) in heterogeneous IP-networks, even if the change is not planned. The latter will occur in the case of a link breakage. Maintaining the call is the primary goal, while making the handover seamless is secondary.

Throughout this thesis, the term "session" is used, since this is the correct term in a Session Initiation Protocol (SIP) context. SIP supports different sessions, including voice, video and others, but sessions other than voice calls are out of scope for this thesis.

By seamless in this context, I mean a total handover time not exceeding 200ms[7] for planned handover. For link breakage a longer, not time specified, handover time is acceptable as long as the call is not dropped.

The motivation of addressing this issue, is a wish to utilize some or all wireless carriers available to a terminal (WLAN, WiMAX and others, ergo heterogeneous networks) on the same session, to be able to maintain the session even if one of the carriers becomes overloaded or breaks. Many proposals have been made to address the issue of VoIP handover, but currently no one seems to have achieved seamless handover for *real-time* applications, nor managing link breakage.

The difference in handling mobility in a homogeneous system like GSM, and in heterogeneous networks, is that the latter lack a common mobility management system. During a change of network interface on a terminal, it is not possible to maintain the session since the session is associated with an IP-address on an interface. The session will then have to be re-initiated, using the new interface.

1.3 Methods

The methods used in this thesis include a literature study to get an overview of known mobility methods, and to decide which method is the better alternative to use to achieve the goal stated in section 1.2. Parts of the work are done in collaboration with Elin Sundby Boysen. Boysen is a Ph.D candidate at the Norwegian Defence Research Establishment (FFI) and UniK - University Graduate Center. Protocols and devices has been modified to make up a test-scenario, and show a "proof of concept".

1.4 Outline of the thesis

Chapter 1: Introduction to the thesis.
Chapter 2: Background describes the background for the research area.
Chapter 3: Approaches to mobility contains more details about mobility.
Chapter 4: Proactive handover describes the Proactive Handover Solution.
Chapter 5: Implementation describes the implementation of the Proactive Handover Solution.
Chapter 6: Results and Evaluation presents and evaluates the results of the experiments.
Chapter 7: Conclusion and proposals to further research.
Appendix A contains the source code.
Appendix B contains the measuring methods.
Appendix C contains the SIP message format
Appendix D contains the Wireshark-traces (sources for measured times)

1.5 Reading the thesis

It will be assumed that the reader has basic knowledge about the ISO/OSI reference model, IP-networks, including protocols such as TCP and UDP, HTTP, and about mobile communication systems like GSM/UMTS and equivalents.

Background information about these subjects can be found in Schiller[3] chapter 2.2(TCP and UDP), Leon-Garcia[8] chapter 8.5(ISO/OSI) and 2.5(IP-networks) and in Schiller[3] chapter 4.1(GSM/UMTS).

It is recommended that the chapters are read in the order that they are presented, but readers with good knowledge to mobility and VoIP might skip chapter 2, and continue reading in chapter 3.

Necessary background information beyond these topics will be provided.

Chapter 2

Background

The demand for bandwidth on mobile terminals is increasing, and the support for using heterogeneous networks is essential in that context. In addition, after a period of using Voice over IP (VoIP) mainly as a substitute for *fixed* telephony like ISDN, we see a wish for using VoIP even on mobile terminals.

This chapter describes basic background about heterogeneous networks, mobility and Voice over IP.

2.1 Heterogeneous networks

The expression "Heterogeneous Networks" means networks of different types. These networks can be WLAN, WiMAX, GSM/UMTS or others. Different networks have different properties, like bandwidth, Quality of Service(QoS), cost, support for stationary or mobile nodes at different velocities, how energy consuming they are etc. Figure 2.1 shows the trade-off between mobility and bandwidth.

For mobile devices it is attractive to be able to utilize the network that is the most adequate for the current session. When downloading large amounts of data at a relatively stationary position, WLAN would be the better carrier. Performing a voice call when the terminal is running low on battery, would favour GSM as the carrier in question. When it comes to handover between different networks, this is a challenge because heterogeneous networks do not have a common mobility management system.

New services can demand higher bandwidth, lower delay and other qualities, and thus it will not be possible to fulfill all requirements with one single type of carrier. One solution to this challenge, is to use different types of networks, where the best suited for each situation will be used.

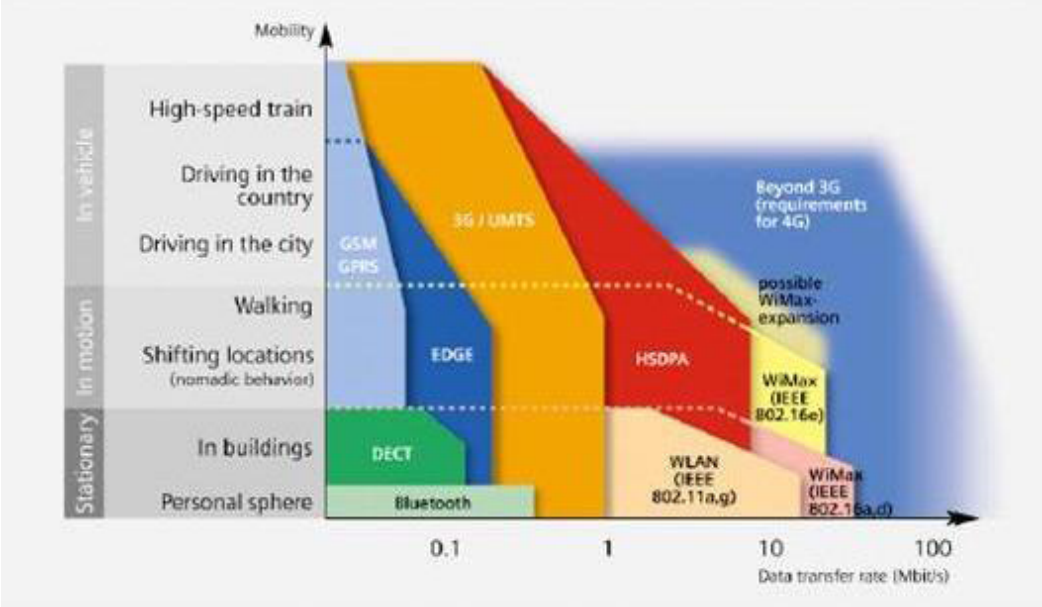

Figure 2.1: Mobility vs. bandwidth

The challenge in this context, is to provide a handover between the different carriers, since there are no common mobility management system present in a heterogeneous network environment.

One large group which address this issue, is the *WINNER project*[9]. WINNER is a group of 41 partners from the communication industry and research institutions which are working toward enhancing the performance of mobile communication systems. This project aims to outdo all current wireless systems, and plans to solve this by creating a network with different "modes", which represents carriers with different qualities. These different carriers will have a common mobility management system which will provide handover between the different "modes".

Different wireless networks have various support for motion of the terminals. The networks can be categorized into 3 velocity classes, which are:

- Stationary
- Pedestrian
- Vehicular

Table 2.1 shows the bit rate and velocity class for different wireless systems.

Name/standard	Max (Download) Bit rate(Mbit/s)	Velocity class
WLAN/802.11g	54,000	Pedestrian
WLAN/802.11n	600,000	Pedestrian
WiMAX/802.16	268,000	Pedestrian
Mobile WiMAX/802.16e	63,000	Vehicular
GSM/GPRS	0,054	Vehicular
UMTS	0,384,000	Vehicular
HSDPA	14,400	Vehicular

Table 2.1: Different wireless systems[3][4]

2.2 Mobility

The level of mobility supported by different systems is often divided into four categories[6]. These are:

- Personal Mobility
- Service Mobility
- Session Mobility
- Terminal Mobility

Personal Mobility refers to the ability of a user to access telecommunication services from any terminal on a basis of a personal identifier, from anywhere and at any time. This includes the networks' capability to locate the terminal for the purposes of addressing, routing and charging.

Service Mobility refers to the ability of the network to provide the user with personalized services, with the expected QoS, independent of the user's location. It also allows users to maintain their services while in motion and independent of Network Access Point.

Session Mobility refers to the user's ability to maintain an active session while switching between terminals.

Terminal Mobility refers to the ability of a terminal to, while in motion, access telecommunication services from different locations and provide the same services.

Current VoIP based on the Session Initiation Protocol (SIP) would be placed in the "Session Mobility" category, and implicit the previous categories of mobility is also covered[6]. This implies that the user is able to access the service

anywhere, with any terminal, using different Network Access Points, and even maintain an active session while switching between terminals.

This can be illustrated by an example. Consider a scenario where a worker in an office is walking around, having an active SIP session on his wireless phone. When he returns to his desk, he wants to continue the call from his wired phone. This is the concept of session mobility. What does *not* work, is a change of NAP on the same terminal. This can be illustrated by a scenario where the worker walks out of the building; beyond the coverage of the Access Point he is connected to. Even if he is covered by several other Access Points, the session will break.

When extending VoIP to mobile devices, the initial mobility support in SIP is not sufficient. Mobile users demand a mobility support equivalent to existing cellular systems or better, that is, seamless change of NAP. This principal is known as seamless handover or handoff [3]. In addition, the user should be able to use the terminal while in motion at different speeds. In a next-generation telecommunication system, the terminal should be able to operate even at speeds beyond 250km/h which is the maximum for GSM. ([3], page 117)

Support is wanted not only for planned/controlled handovers, but to be able to maintain a session even during link-breakage, using other carriers available.

2.2.1 Handover

The expression handover means the procedure of changing NAP. The expression handoff is also commonly used, and describes the same function, but throughout this thesis, the expression handover will be used. Different types of handover are often categorized by the way they affect the packet stream.

1. Hard handover
2. Soft handover
3. Seamless handover

Hard handover will result in an interruption of the packet stream, and thus a significant glitch in the speech if the network is used to transfer voice. The soft handover is executed with a minimal packet loss. The glitch can be noticeable, but not as significant as for the hard handover. The new connection is being set up before the existing breaks. When it comes to seamless handover, the user is not supposed to notice the change of NAP, and neither QoS, security or features in the networks are affected[10].

The type of handover might also be categorized by how the handover is conducted.

1. Proactive Handover
2. Reactive Handover

In the first case, the handover is planned, and the new connection is ready before the existing one breaks. In the latter case, the handover procedure is not started before a link breakage is detected.

The last way to categorize handovers, is in respect to whether the handover is executed in a homogeneous or heterogeneous network environment. A heterogeneous network is made up from networks of different type, for example a WLAN and a GSM network. A homogeneous network consists of only Network Attachment Points of the same type.

1. Vertical Handover
2. Horizontal Handover

The expression Vertical Handover is used for homogeneous network environments, that is handover inside one system like handover in the GSM system. Horizontal Handover describes handover in heterogeneous networks, ergo handover between different systems. ([3], chapter 11)

2.3 Voice over IP (VoIP)

Voice over IP, also called Internet Telephony means using a packet switched (PS) IP-network like the Internet to transfer voice calls as an alternative to the traditional circuit switched (CS) networks like ISDN.

VoIP uses different protocols to initiate calls, and to transport the speech. The dominating method is to use the Real-time Transmision Protocol (RTP) for transfering ht voice, and the Session Initiation Protocol (SIP) for the signalling. This thesis focuses on VoIP based on SIP, since it is todays most widespread signalling protocol for VoIP. It is used by most Internet Telephony Service Providers (ITSPs), accepted by the 3GPP as the signalling for the IP Multimedia Subsystem (IMS)[11] and the Generic Access Network (GAN)[12]. It is also used in most commercial mobile VoIP proposals.

Different codecs are used to convert from analog voice to digitally encoded voice, which in turn is being sent over the network using the Realtime Transport Protocol (RTP)[13]. Codecs vary in sound quality, required bandwidth and other qualities, consequently the different codecs have their strengths and weaknesses. A commonly used codec is the G.711[14], which makes a good compromise between sound quality and required bandwidth. Sound quality with G.711 is equivalent to ISDN given that the network quality is good (bandwidth, jitter etc. at acceptable values), due to the same parameters; 8 kHz channel width and 64kbit/s bitrate.

Another codec worth mentioning, is the G.722.2[15] that provides a better sound quality due to 16 kHz channel width. No further details about codecs will be given here, but the interested reader will find more information in the specifications from the International Telecommunication Union (ITU) [16].

Fixed VoIP has in short time reached a relative high penetration rate (Norway, 29 per cent of broadband customers, 2006[17]), and is considered a reasonably priced alternative to the more expensive ISDN or POTS[1][18], especially for long distance calls. In spite of this, few customers and providers utilize the other qualities of VoIP, for example the possibilities for mobility and better sound quality.

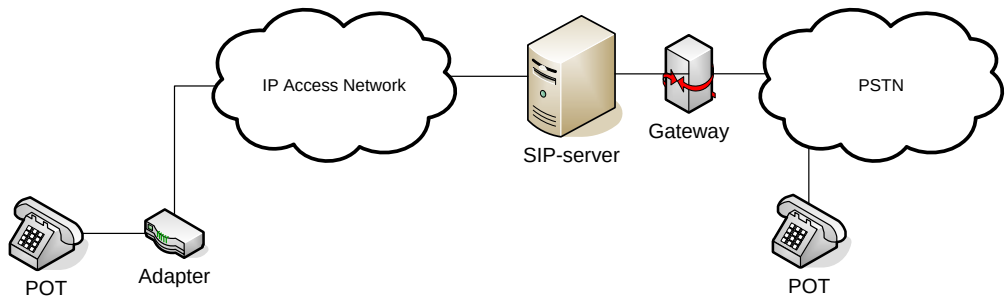

Figure 2.2: Typical VoIP usage

2.3.1 Session Initiation Protocol (SIP)

The Session Initiation Protocol (SIP) is an application-layer control protocol standardized by the IETF[19]. SIP is used to initiate, modify and terminate multimedia sessions over a network, for example the Internet. Multimedia sessions in this context mean voice calls, video-conferences and/or instant messaging

[1]POTS is a term used to describe analog telephone.

(IM).

The protocol is standardized by IETF, and is easily implemented with other IETF protocols like TCP, UDP and RTP. SIP has in short time gained widespread acceptance. SIP is very similar to HTTP; it is text-based and thus easily readable to humans and much of the message header syntax and many HTTP codes are re-used (for example "404/Not Found"). SIP is transport-independent and can run over TCP, UDP, ATM and other.

SIP is described as a peer to peer protocol, and should work without any intervening infrastructure. Proxy and Registrar network elements are however required for SIP to work as a practical service. SIP-Proxy servers route requests to the users' current location, authenticate and authorize users for services and provide features to users. ([8] chapter 10.7)

A SIP-Registrar server is a special kind of SIP-server, which might be compared to the Home Location Register (HLR [3], chapter 5.5) in GSM. The SIP-Registrar is a database containing authorized users, and their current location (IP-address). It is important to notice that the distinction between the types of SIP servers is logical and not necessarily physical.

When initiating a SIP session, SIP proxy servers are used to forward SIP requests to the receiver, while Registrar servers assist the SIP-proxies in obtaining the users' current location. After call setup, the packages (media flow) traverse the network directly between the call participants, without any intervening servers, as seen in figure 2.3.

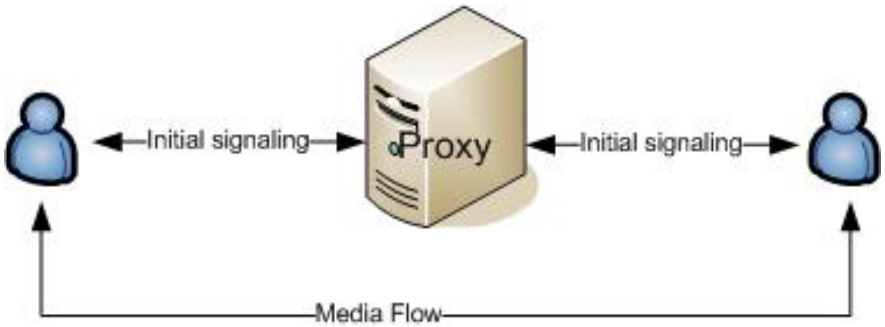

Figure 2.3: Standard SIP with proxy

It is important to notice that even though SIP us used to set up the session, the media flow is independent of SIP. Consequently the signalling and the media flow do not necessarily traverse the network using the same path.

Even though traffic can be transferred directly between participants during a call, this is not always the best solution for practical purposes. The alternative is to use a "Back to Back User Agent" (B2BUA). This is a SIP-server which operates as two User Agents back to back. When a B2BUA receives a call, it accepts the call (like a User Agent Server), and initiates a *new* call (like a User Agent Client) to the desired receiver. This setup is shown in figure 2.4.

Figure 2.4: Back to Back User Agent

This setup has its advantages over the ordinary request-forwarding. A B2BUA can perform media-bridging for the purpose of the call participants using different media codecs and even different transport protocols. The B2BUA is also in control of the media stream, which can be usefull in a large number of applications. The B2BUA might be compared to a PBX[2] in traditional telecommunication.

The protocol uses an offer-and-reply model in call setup. The unit initiating the call offers a set of media attributes which include capabilities (like video/not video), media encodings, transport protocol and others. The corresponding unit responds to this request with a subset of media attributes, which are the ones that both participants support. They are listed in preferred order, consequently the first codec and the first transport protocol listed will be chosen et cetera. SIP uses the Session Description Protocol (SDP)[20], for this purpose.

The end-points are in a SIP-context called User Agents. The User Agent consists of a User Agent Client (UAC) and a User Agent Server (UAS), which initiates and replies to requests respectively.

2.3.1.1 Session Description Protocol (SDP)

The Session Description Protocol (SDP)[20] is an IETF protocol, used to describe sessions with sufficient information to discover and participate in a multimedia

[2]PBX stands for Private Branch eXchange and is a telephone exchange that serves a company or office.

session. SDP describes multimedia sessions for the purpose of session announcement, session invitation and other forms of multimedia session initiation. SDP is enclosed in the SIP INVITE requests and used in the offer and reply model for the purpose of call participants to agree on media attributes for the session.

2.3.2 Other VoIP signaling protocols

There are protocols that can be used as alternatives to SIP. One of them is the H.323 protocol [21], which was standardized by the ITU in 1996, and was the first protocol that used the IETF Realtime Transport Protocol (RTP)[13] to transport audio and video over IP networks. Lately, H.323 has lost most of its position to SIP, but is still used in many applications.

Another signalling protocol is IAX[22], which initially was designed for communication between Asterisk PBX's[23], but is growing in popularity even for clients due to its abilities to traverse NAT[24]. Problems related to NAT traversal are out of scope for this thesis.

In addition, there is a high amount of proprietary VoIP protocols used in software and hardware VoIP products from different vendors.

Chapter 3

Approaches to Mobility

Many proposals have been made to Voice over IP (VoIP) mobility, but most research activities have been focused on the network layer in the OSI stack, that is Mobile IP (MIP) and its different variants. Lately, much work has been done on higher-layer solutions, due to their more flexible and portable qualities. This chapter will present an overview and evaluation of the most commonly accepted and widespread solutions that are available. There are two dominating ways to introduce VoIP mobility, GAN and SIP.

3.1 Generic Access Network (GAN)

3.1.1 Overview

The Generic Access Network (GAN) project was initially called Unlicensed Mobile Access (UMA), and the project was founded by a number of operators and vendors of mobile communications [25]. Their goal was to extend GSM/GPRS services to work over unlicensed spectrum[26]; namely WLAN[27] or Bluetooth[28]. The initial specification was published in September 2004.

In May 2005, the 3rd Generation Partnership Project adopted the project, and named it Generic Access Network (GAN), indicating that not only WLAN/Bluetooth, but even wireless technologies operating on *licensed* bands, will be supported. In spite of this, current documentation only specify WLAN access in addition to GSM/GPRS, but states that *WLAN technologies other than those compliant with IEEE 802.11 1999, such as HiperLAN or Bluetooth, are not described specifically in this version of the present document. However, they are not excluded.*([29], page 9)

The motivation for this project is evident from an operators point of view. In the current GSM/UMTS networks, the radio resources are limited. In addition,

due to smaller coverage area, the UMTS deployment is a highly expensive matter, and for that reason the current UMTS penetration is limited. For service providers, GAN might be the solution to increase coverage area and bandwidth, and release valuable radio resources in the GSM/UMTS RAN. The main reason for customers to use GAN, is increased bandwidth and lower cost on calls initiated from their home or office. Nevertheless customers might question why they are charged for data which is going over their own IP-connection, which they are already paying for.

The basic GAN architecture is shown in Figure 3.1.

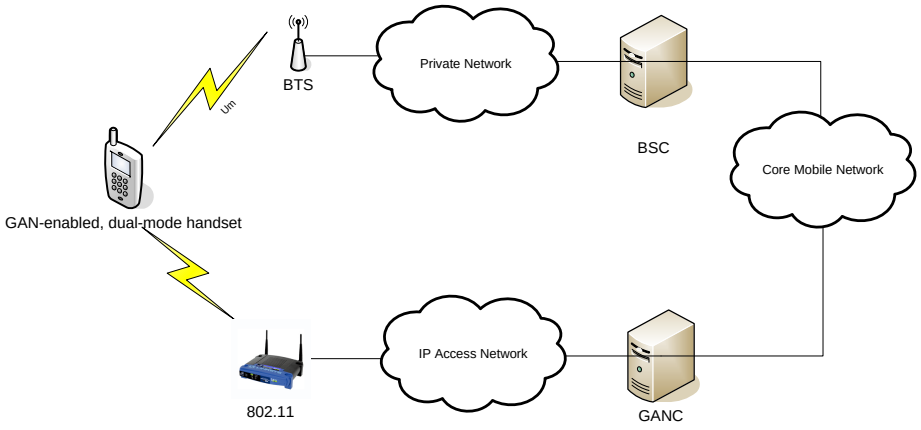

Figure 3.1: GAN architecture

3.1.2 Technology

GAN introduces a new network element in the operators' network, namely the Generic Access Network Controller (GANC). This unit offers GSM/GPRS services to the handset over an IP-access network, and is treated like a Base Station

Controller (BSC)[1][3] by the core mobile network. The handset has to be dual or multi-mode, in other words capable of using other radio access technologies such as WLAN in addition to GSM, and include the needed GAN-software.

The GAN operation includes seamless roaming between GSM and other access-network, and offers the same services, with equal quality and security, as GSM Radio Access Network (RAN). This section will describe GAN technology in more detail.

3.1.2.1 Network elements and interfaces

GSM is one of the technologies that have made great success with open interfaces. When an interface is precisely described between logical entities, the use and interoperability of devices from different vendors is no longer a problem. This makes the deployment task for operators easier.

This concept is adopted by GAN as well. It enables an easy deployment of GAN technology for mobile operators, and vendors of handsets can design their products without deep knowledge about the operation of the GANC.

The new GANC element introduces one new interface called *Up*, which is the interface between the Handset and the GANC. Toward the core mobile network[3], the GANC use the already defined interfaces *A* and *Gb* for communication with the Mobile Switching Center (MSC) and the Serving GPRS Support Node (SGSN) respectively. The MSC handles the voice, while SGSN handles the data traffic. The network elements and interfaces are shown in figure 3.2.

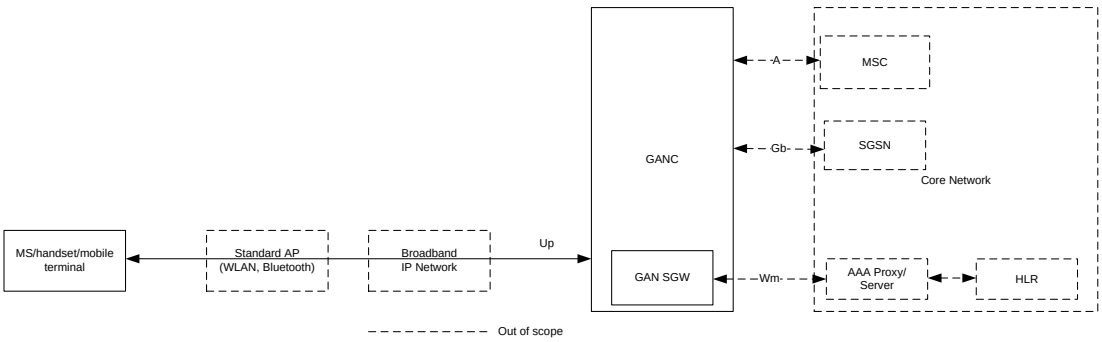

Figure 3.2: GAN functional architecture[1]

[1]The BSC controls the radio signals of one or multiple cell sites, and performs radio signal management.

3.1.2.2 GAN Operation

When the handset is in GSM-coverage only, it operates like an ordinary GSM cellular phone. When entering the WLAN zone at home[2], initial WLAN authentication is executed, and an IPsec-tunnel[30] between the handset and the GANC is established. This ensures a secure connection toward the GANC. Authentication toward the core mobile network is then performed.

From that point on, SIP is used for signalling between the handset and the GANC, and the GANC converts the signalling and payload to appear like data from a BSC and then communicates this to the core network using the *A* and *Gb* interfaces, for voice and data traffic respectively.

Since the GANC is seen as a BSC by the core mobile network, handover between WLAN and GSM is carried out in the same way as a intra-BSC handover[3].

3.2 SIP-based solutions

3.2.1 Overview

The various SIP solutions use, as the name implies, the SIP-standard. SIP is easy to modify and usable for various purposes, thus the difference in the various solutions lies in how they use SIP, and which technologies they use SIP in cooperation with.

SIP initially supports Session Mobility by using re-INVITE[4] methods, but do not support change of wireless carrier and thereby interface on the same terminal in a heterogeneous network environment. In homogeneous mobile systems like GSM, mobility support was taken into consideration already at the design stadium, while heterogeneous systems lack a common mobility management system. Different proposals have been made to address this problem.

3.2.2 SIP in conjunction with GSM

This solution aims at the same as GAN, but is less integrated to the core mobile network, and thus more attractive for virtual operators, that is operators which do not own their own infrastructure. Several operators are using this solution, for example *Hello*[32] in Norway. This is not a standardized solution like GAN, and thus it is implemented by various providers in different forms. This raise

[2]Different implementations support only the WLAN AP at home, or any WLAN AP.

[3]Handover between cells belonging to different BSCs[31].

[4]There are no SIP-messages named re-INVITE. A normal INVITE message is used, but is commonly referred to as re-INVITE when sent subsequent to a ordinary INVITE.

challenges when it comes to interoperability.

Even though both solutions aim for Fixed Mobile Convergence (FMC)[5], and offer access to GSM services over other access technologies such as WLAN, their functional operation and configuration are very different.

The basic architecture for this solution is shown in figure 3.3.

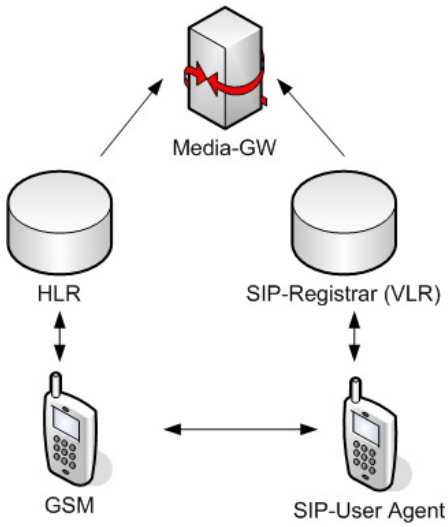

Figure 3.3: SIP in conjunction with GSM

The solution is based on a GSM cellular phone with a SIP User Agent as a central part of the software. When the terminal is in WLAN coverage, the SIP UA is used, otherwise it operates as an ordinary GSM cellular phone.

The core network in this solution consists of a normal GSM core network, and one or more SIP-servers. In addition, a media gateway is needed to provide media re-coding for VoIP calls to PSTN/GSM. The SIP-Registrar can be regarded as a Visitor Location Register (VLR)[6], since the terminal must be registered in the SIP-Registrar server when not using GSM.

Using VoIP when in WLAN coverage and using GSM when not in WLAN coverage is a minor subject to address. The challenge in this context is to provide handover between GSM and VoIP, and to change carrier without user interaction. To address these problems, the terminal will have to measure WLAN signal

[5]Convergence between cellular and fixed telephony; provide both services with a single phone.

[6]The database containing temporary subscriber information in GSM.

strength and initiate a GSM call before loosing WLAN coverage. While maintaining two simultaneous calls, it must make a handover to GSM. The same procedures applies to GSM to WLAN handover, except for WLAN need to be the preferred carrier.

In addition, for situations where the client is idle in WLAN coverage and suddenly loosing the WLAN coverage, the terminal should be able to de-register from the SIP-server using GSM. If this is not handled, the client may be registered on the SIP-server, and incoming calls would be routed to WLAN instead of GSM in a long period after the client left WLAN coverage.

All services, like SMS and MMS need to be handled in this solution as well. There are two ways of providing this support. The simplest way is to deliver SMS and MMS over the GSM network. SMS is carried of the associated control channel which is a dedicated GSM-control channel(600bit/s). MMS is carried by the GPRS data channel to the WAP server. This solution will require that the handset uses both GSM and WLAN simultaneously. The more complicated, but better solution is to deliver SMS and MMS over the IP connection. This solution will comprise an SMS/MMS gateway in the core network. The SMS will then be sent over the IP connection using SIP, and then the message will be converted to a normal SMS in the handset.

3.2.3 Other SIP Mobility proposals

Several proposals have been made to extend the mobility support in SIP. One of the first proposals was made by Wedlund and Schulzrinne in "Mobility Support using SIP"[33]. The solution was further elaborated in "Application-Layer Mobility using SIP"[34]. In the proposed solution, the terminal sends a new INVITE request with its new IP-address, and updated description of the session to the callee. As soon as the callee receives this request, it will start sending the data to the new location. Then the terminal will send a new REGISTER request to its home registrar-server. This procedure is shown in figure 3.4. This solution will generate a handover delay. The time of the delay will depend on the time consumed to send and process the INVITE request, and the time it takes to detect the link-breakage and retrieve new IP-address.

Address acquisition is recognized as the main contributor to the handover delay by Chahbour et. al. In the article "Fast Handoff for Hierarchical Mobile SIP Networks"[7] they propose a "Predictive Address Reservation" combined with a hierarchical architecture to reduce the handover delay. In this way a new IP-address is obtained and registered before the handover takes place. This will reduce the handover time, but can not guarantee a seamless handover.

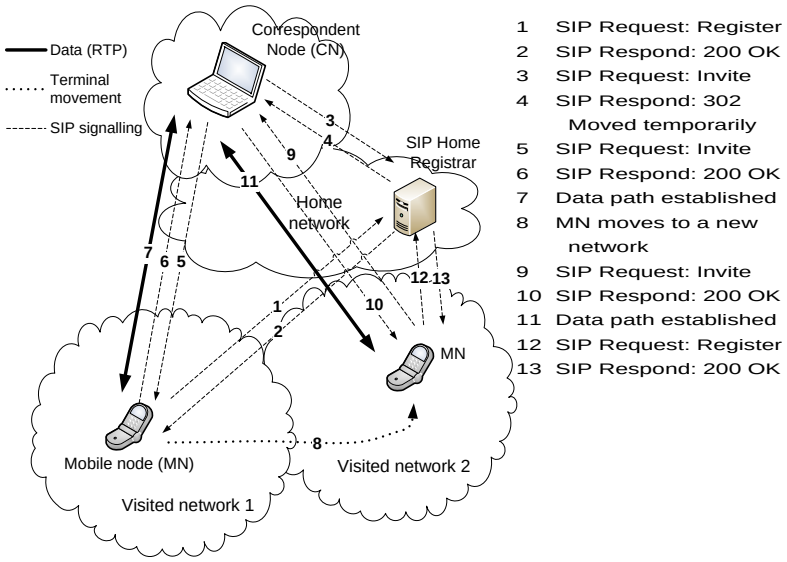

Figure 3.4: Handover Wedlund [2]

Banerjee et. al. propose in the article "SIP-based Mobility Architecture for Next Generation Wireless Networks"[35] a solution for vertical soft handover. The terminal has different network interfaces that can be used simultaneously. Every domain it is supposed to communicate with, will include a SIP B2BUA and a media gateway which perform RTP-package forwarding, duplication and filtering. The terminal initiates the handover, which involves RTP-packet duplication and transmission over both interfaces in the old B2BUA. The terminal does packet filtering, and the old and new base station communicates with each other. This is shown in figure 3.5.

When the handover procedure is finished, the terminal will send a REGISTER request to the home registrar for the purpose of updating its location. This solution can prevent packet loss and delay, but the architecture significantly increase the complexity, due to the requirement for SIP B2BUA's in all access point.

Paolo Bellavista et. al proposes in the article "SIP-based Mobility Architecture for Next Generation Wireless Networks"[36], a solution comprising an application-layer middleware. By using buffering and "Handover Agents" in each visited subnet, this proposal achieves session continuity. The packages are being buffered in both the old and the new domain, which ensures that no packages are lost during handover. In this way zero packet loss can be guaranteed, but

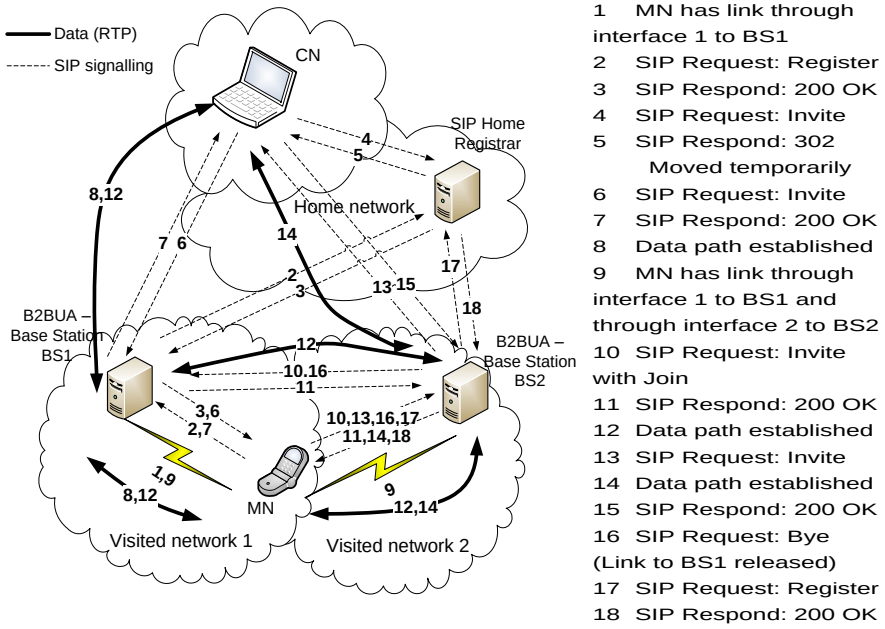

Figure 3.5: Handover Banerjee [2]

due to the buffering technique used, at the cost of delay. Hence, this solution is not suitable for real-time communication. It also require a Handover Agent in each subnet, increasing the complexity of the proposal. The solution is shown in figure 3.6.

These papers points out the need to address the problem of packet loss during handover, and that this might be solved by the use of packet duplication and -filtering. The use of a B2BUA also stands out as a good solution. The problem of managing link breakage in a heterogeneous network environment, is not addressed.

3.3 Mobile IP (MIP)

Mobile IP (MIP) [37] allows portable devices to move from one area to another while maintaining communication sessions. The four basic entities in the Mobile IP architecture are:

- Mobile Node (MN)
- Home Agent (HA)
- Foreign Agent (FA)

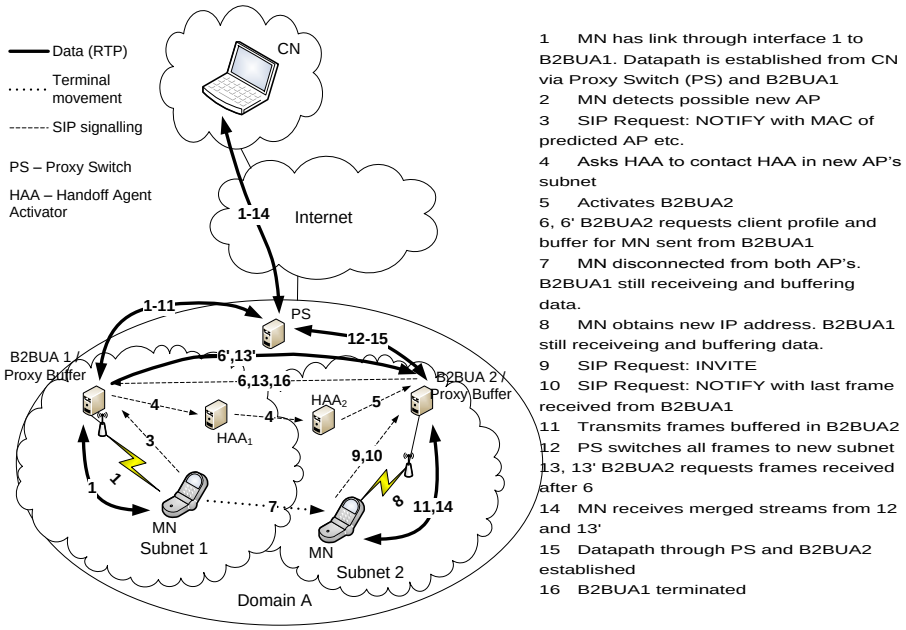

Figure 3.6: Handover Bellavista [2]

- Correspondent Node (CN)

The Mobile IP architecture is shown in figure 3.7.

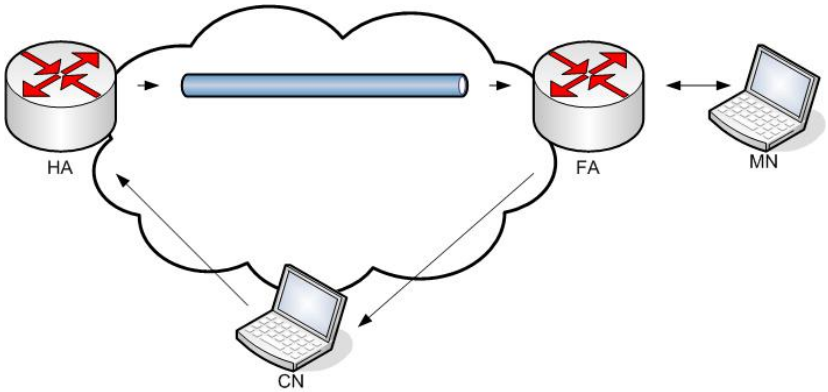

Figure 3.7: Mobile IP

The Mobile Node (MN) is the unit which is in motion, while the Correspondent Node (CN) is the unit it communicates with. In the MNs home network, there is a Home Agent (HA) and in the visited network there is a Foreign Agent. The data packets arrive the home network via ordinary IP routing. The package is intercepted by the HA, and tunnelled to the FA. This tunnelling involves a new

IP header, which can be called a outer IP header. This is shown in figure 3.8[8].

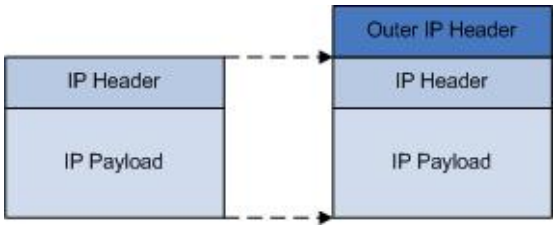

Figure 3.8: IP encapsulation in MIP

The FA de-tunnels the package, and delivers it to the MN. Packages from the MN to the CN are routed using ordinary IP-routing. This is known as triangle-routing.

3.4 Mobility Methods Comparison

Mobile IP is a good solution for several applications using TCP/IP, like Web-browsing, mail and file-transfer, but introduces a handover delay that far exceeds what is acceptable for real-time applications like voice.

GAN is a comprehensive specification defining the extension of GSM networks using WLAN. It is easily deployed, and integrates closely with existing GSM networks. GAN is really nothing but GSM over IP, with all the advantages and disadvantages this implies. GAN will offer security that is equal, or better compared to GSM, due to the use of IPsec tunnel between the handset and the GAN controller. The handover support in GAN is limited to GSM-WLAN or WLAN-GSM, while handover between WLANs or other IP-networks is unsupported. Thus, for session mobility in heterogeneous networks GAN is too closely tied to GSM.

The issue of seamless handover is best solved at the application layer. There are several reasons for this. First, it is desirable to have the possibility to make local solutions. By this I mean for example to modify only an endpoint and a corresponding server. If the IP stack is to be modified, this must be done everywhere.

SIP is also a very flexible protocol, with great support for making extensions, ability to use a wide range of codecs and runs over any IP network. However, the initial mobility support in SIP is not sufficient. To achieve a seamless handover, "make before break" is needed.

The SIP-solutions proposed by Banerjee, Bellavista and others are either based on buffering of packets, and thus best suited for streaming applications, or comprise a very complex architecture.

3.4.1 Choice of mobility solution

As shown in section 3.4, the different mobility methods have their advantages and drawbacks. In my assessment, SIP is the most flexible and best suited basis for making a seamless handover, and to maintain sessions during link breakage.

Due to its network independent quality, and also that it is independent of operator, a SIP-based solution will work over any present or future IP-network, and thus should be well suited for a heterogeneous network environment. Another important aspect is the ability to make a local solution. For these reasons, the planned experiment/test setup will be based on SIP.

Chapter 4

SIP-based Proactive Handover

Based on the background in chapter 2 and the assessment of different mobility solutions in chapter 3, I have in collaboration with Ph.D candidate Elin Sundby Boysen[1] made a proposal which we would like to call a "SIP-based Proactive Handover". This chapter describes the solution.

4.1 Introduction

Handover was described in section 2.2.1, and is a generic expression for the change of Network Attachment Point (NAP). A NAP is the connection point to the network, and can be an 802.11 Access Point, a GSM base station or others.

In SIP mobility, we differ between pre-call and mid-call mobility. The first one is handled thorough re-REGISTER at the server when changing NAP/IP-address. Mid-call mobility is more difficult to manage, and is the focus of this thesis. This solution was designed with focus on handling link-breakage, but can with slight modifications support seamless handover when the link quality is gradually degraded.

For the case of SIP, the handover is usually conducted in a reactive manner, which means action is taken before it is necessary. The time to make a handover during a link breakage will therefore be given by the time it takes to detect the link breakage, retrieve new IP-address, register the new IP-address and re-initiate the call:

$$T_{Handover} = T_{BreakageDetection} + T_{RetrievingNewIP} + T_{REGISTER} + T_{INVITE}$$

[1]Elin Sundby Boysen is a Ph.D candidate at the Norwegian Defence Research Establishment(FFI) and UniK - University Graduate Center at Kjeller, Norway.

In the proposed solution, the handover is done proactively, which means that actions are being taken before the actual handover takes place. With this solution, the handover time will be reduced to the time it takes to send the re-INVITE. A re-INVITE consumes significantly less time to process than an ordinary INVITE, because all authentication and media negotiation is already done.

As described in section 2.2, the handover is usually planned based on signal levels, load on base stations etc., but when it comes to handling link breakage instead of graceful degradation, this might not be possible.

This solution was designed with the intention to use different wireless IP-networks such as WiFi and WiMAX, but for the experiments WLAN and Ethernet will be used for simplicity reasons. Nevertheless it is expected that the results will be applicable for all networks capable of carrying IP, since all actions is done at layers above IP.

4.2 Solution overview

We presume a terminal with more than one network interface. The type of interface is irrelevant in this context. If more than one network interface can be used, one of them is chosen as the main/primary interface, while the other(s) will act as backup interface(s). The terminal will decide whether it needs more than one interface acting as backup.

Adding this functionality implicates modifying of the SIP-standard, and requires both the caller and callee's equipment to be modified. We can not assume that all end-point will be modified, and in addition we want the proactive functionality to be available even when calling non-SIP clients, like calls to the PSTN/GSM. For that reason, both one client and one SIP-server was modified.

The SIP server and home registrar are implemented on a B2BUA (described in section 2.3.1), which bridges call between the terminal and the Correspondent Node (CN), and thus is in control of the media stream.

When the terminal performs the initial registration toward the home registrar, it registers all its network interfaces, and chooses a priority for each of them. The priority for each interface is signalled to the server by adding the parameter if_q to the Contact Header in the REGISTER requests.

To specify that the different registrations in fact are from different network interfaces on the same unit, and not a location update (change of IP-address)caused by movement of the user, the parameter ua_id is added to the Contact Header

as well. This parameter is a random number generated by the User Agent at the time of the first registration, and kept as long as the registration is valid.

When a REGISTER request is received by the server it will include the parameter *if_no* in the *200/OK* response, to inform the User Agent about the number of registered interfaces. If the client is registering on a server which does not support the Proactive Handover, the additional parameters in the REGISTER request will be ignored, and the *if_no* parameter will not be returned. This informs the User Agent (UA) that it should not send multiple registrations, since these will overwrite the existing registration. This will happen because the server will read this as a location update.

When the UA wishes to initiate a new session, it sends an INVITE request using the primary interface. As soon as the callee answers the call, the User Agent sends a new INVITE with the same Call-ID, using the first backup interface. The B2BUA server perceive that this Call-ID is equal to the one on the active session, and that the request is marked with *a=sendonly* in SDP, and put the new session directly on hold.

To avoid time-out on the passive calls on the backup interfaces, the INVITE requests are periodical resent, but no RTP packets are sent. The solution is shown in figure 4.1. A and A' is two interfaces on the same terminal, while X is the B2BUA-server.

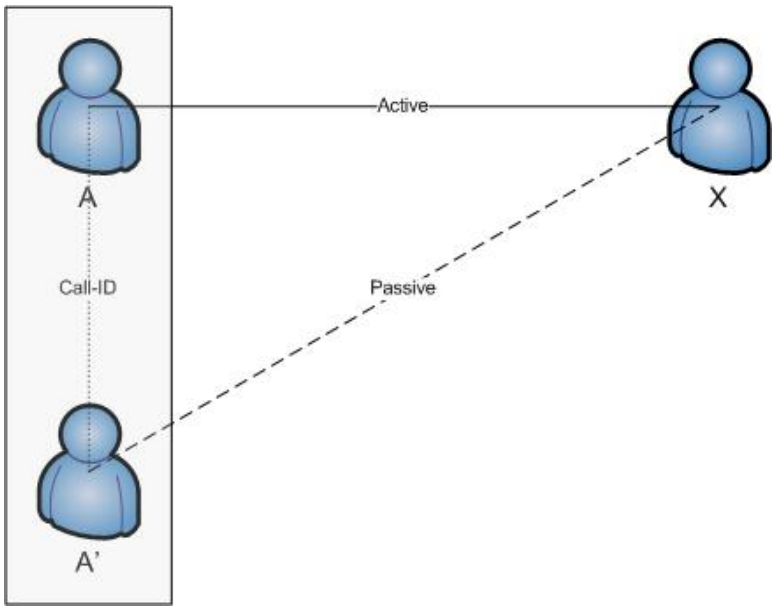

Figure 4.1: Solution overview

4.2.1 Handover Procedure

The handover will be initiated due to either a graceful degradation of the link, or a sudden link breakage, that is, not planned handover. Both situations will require a middleware which informs the User Agent about the link condition. The link quality can be based on different parameters, such as package loss, signal strength information or other.

In the first case, a re-INVITE is sent when the signal level go below a given threshold, using the first backup interface. This is shown in figure 4.2. Through the SDP the B2BUA server is informed that a handover is initiated, and that the RTP-packets are to be duplicated for a given time. The original path is kept open, and the packet stream is sent over both network interfaces. This presupposes packet duplicating capabilities in the server and packet filtering capabilities in the client. The User Agent will then have the possibility to synchronize the two packet streams before the first path breaks, and thus achieve a perfect seamless handover.

We have designed the solution in the way that the client makes all decisions regarding handover, assumed that the terminal itself have the best knowledge about link qualities and available networks.

When the handover is provoked due to a sudden link breakage, the terminal will send a re-INVITE over the first backup interface as soon as the link breakage is detected, and the call/session will continue immediately. The server will then change the path for the media stream. In this case, the server is informed through the SDP that a link breakage has occurred, and thus the RTP packages do not need to be duplicated. This is shown in figure 4.3.

Once the handover procedure is completed, the priority of the registered interfaces will be rearranged, in such a way that the main/primary interface becomes the backup interface and vice versa. This procedure is done automatically in both the client and the server, without the need for any signaling. If there is more than one backup interface available, and a different one is more suited to be the primary interface, a new INVITE request will be sent.

This gives the terminal the ability to move in and out of a wireless zone, without the need to re-register backup interfaces. The new backup interface must in either case send a new INVITE with *a=sendonly* to work as a backup interface for the new active interface.

If the handover can be planned, using packet loss or signal level, this solution might well be used to make a seamless handover in a heterogeneous network en-

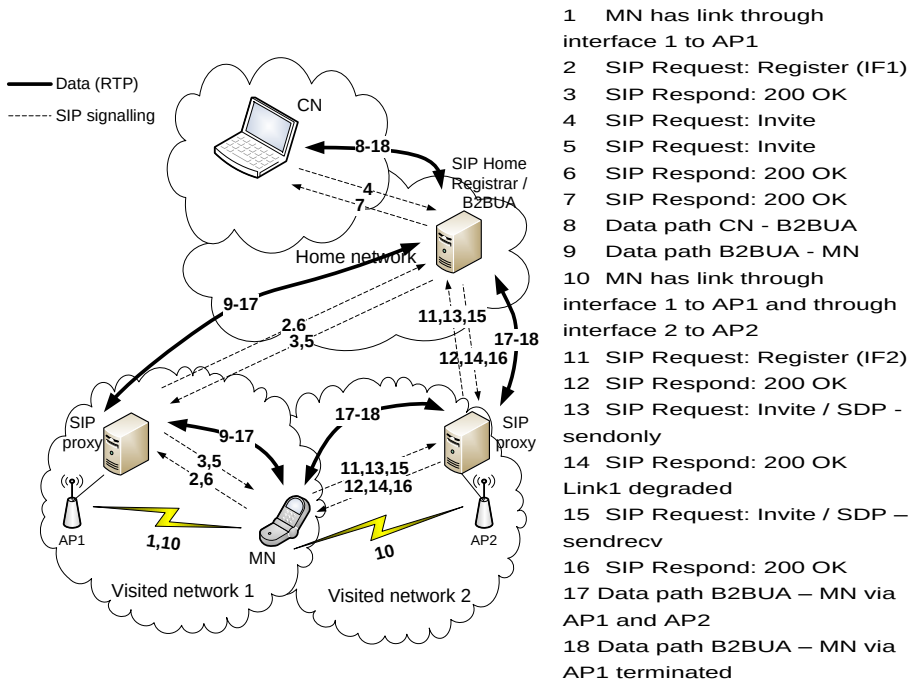

Figure 4.2: Handover scenario with gradual degradation [2]

vironment.

4.3 Technical solution

For the practical experiment, we have chosen to use two SIP-clients (end-points), and a SIP server with B2BUA functionality. The *Asterisk PBX*[23] and the *SIP-Communicator*[38] was chosen as the SIP Back to Back User Agent(B2BUA) and User Agent(UA) to be modified, respectively. The unmodified UA used, is an *X-lite* softphone[39].

The solution is shown in figure 4.4.

This setup has many advantages, but most important for our mission, is that we will be able to implement the Proactive Handover functionality in a controlled environment, that is our SIP server and our SIP client, and the functionality will be available independent of whom the callee is.

Chapter 5 describes the implementation of the Proactive Handover in SIP-Communicator.

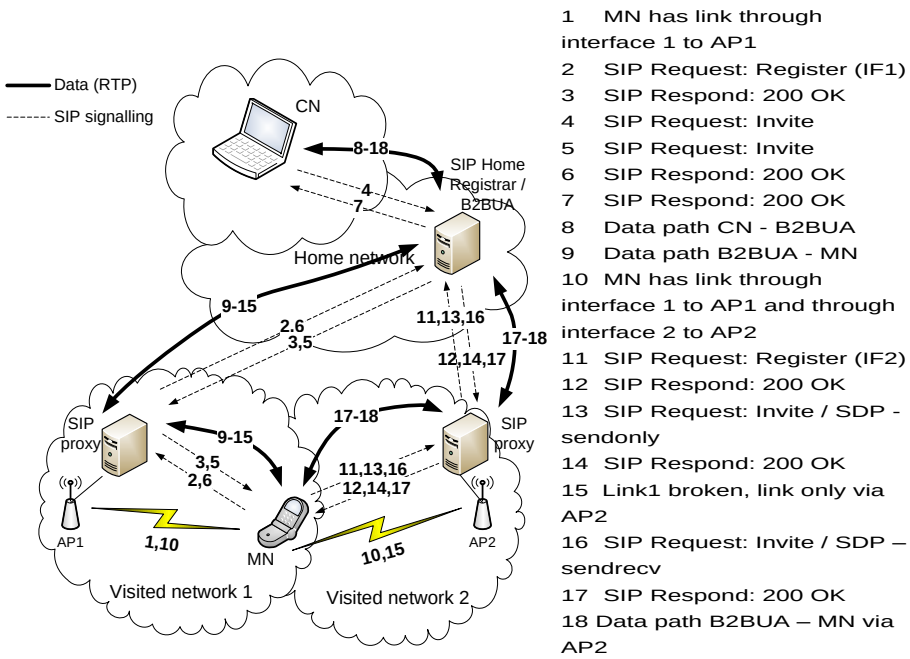

Figure 4.3: Handover scenario with broken primary link [2]

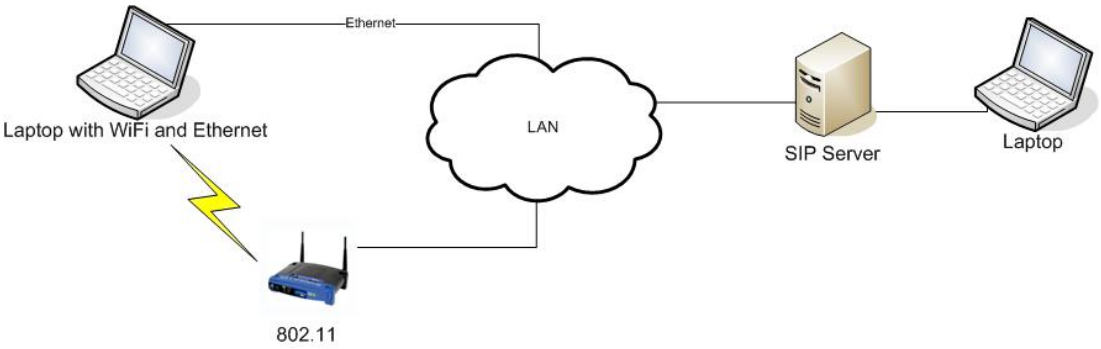

Figure 4.4: Technical solution

4.4 Presumptions and Limitations

To make a good handover, signal strength measuring and link breakage detection is important. As these actions are performed at the link-level, they are not covered here.

The subject for handover in this thesis is heterogeneous wireless networks. For simplicity reasons, the experiment setup will only comprise WLAN and Ethernet. The experiment will in spite of this, have the same validity, because from the application layer, where SIP operates, they are all only IP-networks. SIP is network-transport independent and has no knowledge about the MAC and PHY layers. This will significantly reduce the complexity of the experiment, while keeping the validity.

In this solution, only handovers between networks of different type (known as Vertical Handover) is considered. Because the solution supposes that a new session is set up using another network interface, Horizontal Handover (handover between networks of the same type) will not be supported. However, horizontal handover is generally handled at the link layer.

When changing NAP, we assume that the new network is discovered, and the terminal is authorized to use the network in question. Further we assume that all required Link-Layer authentication is done, and that new IP-address is received from DHCP server or obtained in other ways.

This experiment is done only to show a "proof of concept", and the software is not meant for commercial use in its current form.

It would have been possible to design the solution in such a way that two active simultaneous sessions were used at all times, not only during handover, instead of using one active and one passive session. This would have made it possible to make a completely seamless handover, even during link-breakage. However it was decided against this since two active sessions simultaneously would introduce increased network traffic, higher battery usage on the terminal and more load on the SIP-server.

Further, the solution presume that the carriers in question are able to transport IP-traffic, and that the bandwidth are sufficient.

Chapter 5

Implementation

This chapter describes the implementation of the Proactive Handover solution that was described in chapter 4, for the purpose of making a test setup. The SIP User Agent which will be used is the *SIP Communicator*[38]. The corresponding SIP-server is the *Asterisk PBX*[23], where Elin Sundby Boysen has implemented the solution.

5.1 Overview

For the implementation of the proactive handover solution, we needed to find a SIP client which was open-source, and in addition I wanted to use a program which written in java, due to its portable qualities. The project *SIP-Communicator*[38] was chosen.

The SIP-Communicator is a *java.net* project which in addition to SIP, supports a wide range of protocols to support both audio/video and different instant messaging protocols such as Jabber, MSN, ICQ and others. The project was originally created by Emil Ivov, but now has a large number of developers from all the world.

In the development of the java-code, *Eclipse SDK*[40] was used. Eclipse is an open source Integrated Development Environment (IDE), which can be used with java among other programming languages. In order to be able to retrieve the source code of SIP-Communicator via CVS[1], I joined the project with observer rights. In that way the latest source code was retrieved.

The SIP-Communicator project is under active development, and is neither finished or regarded stable. To avoid problems unrelated to my goals, as much functionality as possible which was not related to SIP, was disabled. Furthermore

[1] CVS is a version control system.

the corresponding server was set to only accept *one* type of codec, namely ulaw[2], due to problems related to media negotiation.

The SIP-Communicator uses the SIP-stack from the *JAIN-SIP project*[41]. JAIN-SIP is a full implementation of RFC 3261, and is also a java.net-project. It was initiated by the *National Institute of Standards and Technology*. Since the implementation of the Proactive Handover requires modification of the standard SIP-messages, the JAIN-SIP stack had to be modified as well. In addition, the *Java Media Framework (JMF)*[42] is used to handle the audio/video, that is capture sound from the microphone and convert it to RTP-packets, and playback the sound in the RTP-packets from the called party.

During the implementation work on the SIP-Communicator, a large number of different problems emerged. As described earlier, the project is not at all finished, and include a high amount of bugs. In addition, the project is based on the OSGI framework, which has contributed to the list of problems as well.

My initial plan was to create two SIP-stacks in the User Agent, where they would do the signalling for one session each. Accordingly one SIP-stack would do the signalling for the active session, and a new SIP-stack would be used for the passive session. Only *one* instance of the media/RTP-handling device (Java Media Framework) would be used. The initial design is shown in figure 5.1.

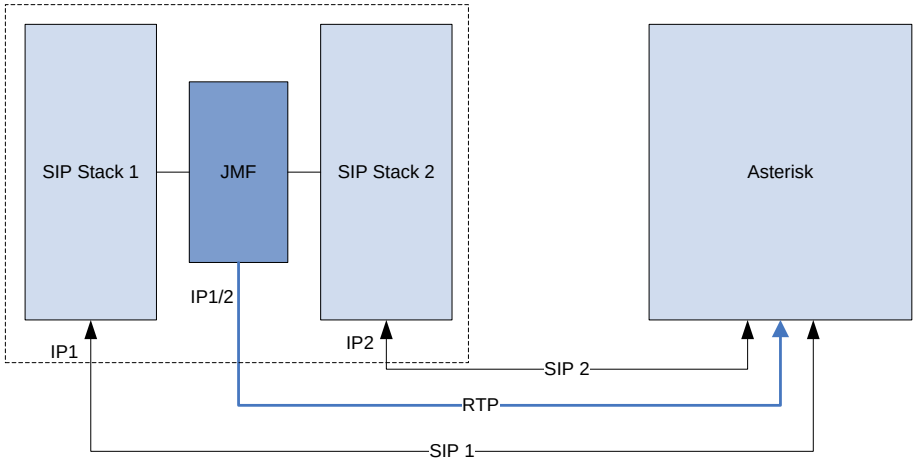

Figure 5.1: 2 SIP Stacks

Due to the high amount of problems which occurred, the planned approach

[2]G.711 μlaw is commonly referred to only as ulaw.

was not possible to accomplish within the given period of time. For that reason, the implementation was slightly changed compared to the solution described in chapter 4. Nevertheless the implementation of the first approach is described in this chapter, since a substantial part of the implementation was already done.

The first step in the implementation was to implement the registration of multiple network interfaces, and then implement the setup of a "passive session" after the first call setup. These topics are described in sections 5.2 and 5.3 respectively. The final implementation is described in section 5.4.

5.2 Multiple registrations

The client starts with obtaining all active interfaces on the terminal, and loopback/localhost[3] interfaces are sorted out. The different interfaces on the terminal is given a value called *q_ if*, indicating the quality of the interface as described in chapter 4. By this I mean in which order the client interfaces should be used when contacting the client, where a lower value is better. This parameter is added to the Contact Header in the REGISTER request. In the testing, a terminal with two network interfaces was used.

In addition, parameters called *ua_ id* and *if_ id* is added, to identify the User Agent and the network interface respectively, as described in the previous section. This is done for the purpose of the server to understand the difference between registration of a newly activated interface on the terminal and a new registration from a different location (location/IP-address update). A new variable is generated every time the application is started.

Since the IP-address of the interface is included in the Contact Header and the Via Header, these has to be generated for each registration. The rest of the headers remain unchanged.

The registration process is then done for all active interfaces, using different values in the headers.

The registration message exchange for two interfaces is shown in figure 5.3.

[3]Interface for the specified by the Internet Protocol (IP), most implementations use 127.0.0.1. Any traffic the host sends to this address, is recieved by the same host.

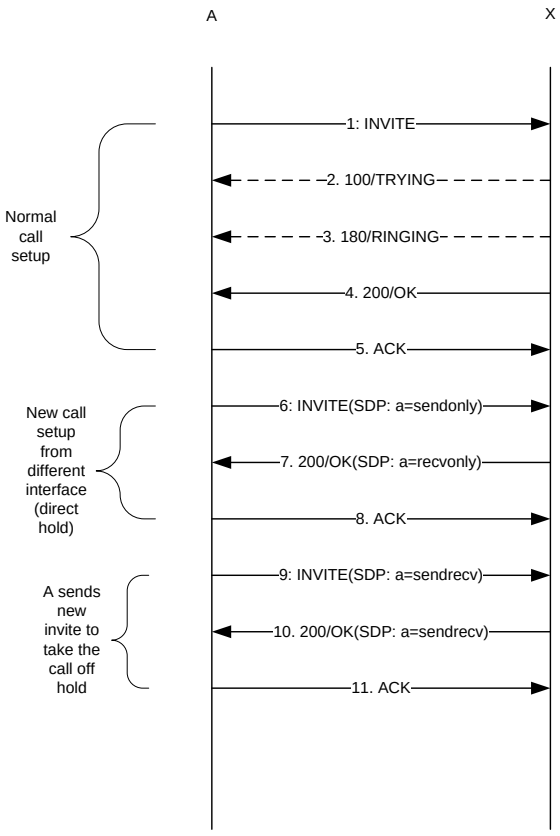

Figure 5.2: Signalling for direct hold

5.3 Call setup

When initiating a call, the client use the IP-address with the lowest *q_ if*, and all headers are generated using this address. The INVITE request include the extra parameters *q_ if*, *ua_ id*, The request is then sent. After the normal, initial signalling, the client receives *200/OK* response, indicating that the callee accepted the call. This call setup is according to the standard.

Since the server is supposed to treat the two simultaneous calls from the client as *one* session, the Call-ID will have to be manipulated. This is done by reusing the Call-ID generated for the first session, when the second session is set up. In this way the server will understand that the second is a backup for the first one, and not a new, independent call.

Since we do not want to set up a backup call before we know that the call will be accepted by the callee, we wait for the *200/OK* response, to confirm that the call is accepted. At this time, a new INVITE request is created, using the IP-address belonging to the the first backup network interface. In this request, the media attribute *a* is set to *sendonly* in SDP, to indicate that the new session is supposed to be set directly on hold.

The server will then respond to this request with *a=recvonly* in the *200/OK* response, to prove that the *a=sendonly* parameter is perceived. The complete message exchange is shown in figure 5.2.

At this time the backup session is set up, but no data (that is, RTP-packets) will be transmitted on the backup session. The INVITE request will however be regularly resent to prevent the backup session from timing out.

5.4 2-Instance Implementation

Due to the different problems with SIP-Communicator, and the close tie between the different elements in the User Agent, it turned out that the planned approach was not possible to accomplish within the given period of time. For that reason the course of this project was slightly changed; Instead of using two SIP-stacks in one instance of SIP-Communicator, two instances of SIP-Communicator will be used. This implementation is shown in figure 5.4.

Even though this seems like a significant change in the project, I believe that this approach will show the principle of the solution in an equivalent way. The largest difference between these two solutions, is that two instances of the media-

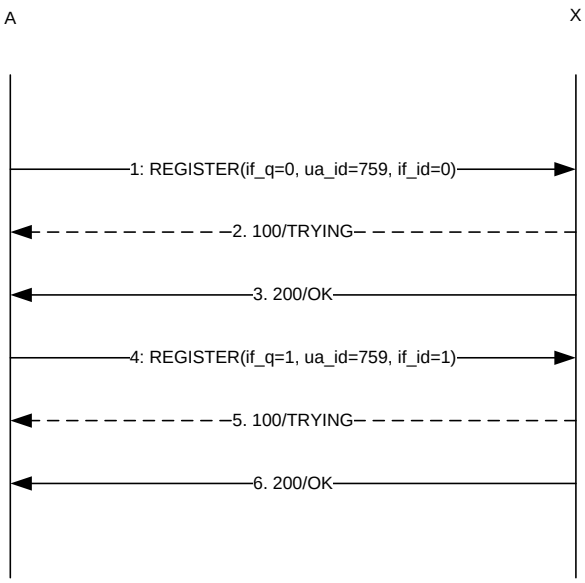

Figure 5.3: Multiple Registrations

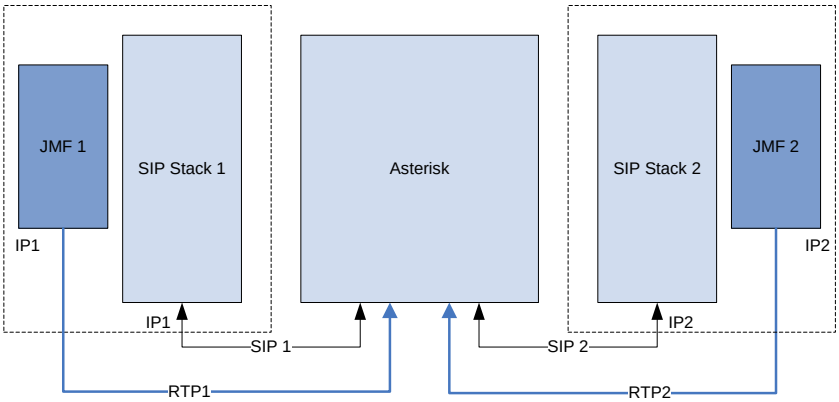

Figure 5.4: 2 SIP-Com. instances

handling device are used, and that the two SIP-stacks do not know of each other, and hence is unable to communicate with each other.

The solution utilizes functionality associated with call-hold, but since this service is not implemented in the SIP-Communicator, everything regarding hold and media handling had to be implemented. For passive sessions, the JMF RTP-handling is not to be started, while it is imperative that it is started for active sessions.

The test setup will be similar to the one first described, but two computers, and hence two instances of the SIP-Communicator will be used. Two sessions will be set up in the same way as described; one active and one passive. The only difference is that these calls are initiated from two different instances of the SIP-Communicator. For the two different calls to be perceived as one in the SIP-server, the Call-IDs have to be equal. For the test setup, the Call-ID is generated from an definite text-string, which is equal in both instances, instead of from the IP-address. The *ua_id* is set equal, while the *if_id* and *q_if* are different.

5.5 Handover Procedure

The handover procedure is initiated by the User Agent with the passive session. A re-INVITE request with *a=sendrcv* in SDP is sent, informing the SIP-server that the call is supposed to be taken off hold. The server will then answer the request with *200/OK* with *a=sendrcv* in SDP, and both the server and the User Agent will start the media stream. In addition to the media parameters, the INVITE request contain the extra parameters *ua_id*, *if_id* and *q_if* as described. The *q_if* parameter is set to *0* in the request, to indicate that the actual interface is now the active one.

When this procedure is finished, the session which started as the active one is put on hold.

Chapter 6

Results and Evaluation

This chapter will present the results from the measurements on the test-setup, and then evaluate the results.

As stated in chapter 4, the total handover time in SIP is given by the time it takes to detect the link breakage, retrieve a new IP-address, make a new registration and make a new call setup.

$$T_{SIP-Handover} = T_{BreakageDetection} + T_{RetrieveNewIP} + T_{REGISTER} + T_{INVITE}$$

The first two variables are independent of the SIP-signalling and handover procedure. The breakage detection is done at the link layer by some kind of middleware, and is out of scope for this thesis. With the Proactive Handover Solution, the time consumed for retrieving and registering the new IP-address is eliminated, since this is already carried out during the set-up of the backup session. The time consumed for processing the INVITE request is minimized, since the decision about media codecs and other parameters is already done.

This section presents the time measurements for a handover with the proposed solution. The method used to find the handover time, is to measure the time consumed from the re-INVITE is sent from the backup interface on the User Agent, until the first RTP-package is received from the server on the same interface. Wireshark Network Protocol Analyzer[43] was used for this purpose.

In the test-setup, the SIP server and the User Agent was located on the same subnet. On a local network, the Round Trip Time(RTT)[1] is neglectable. In the test setup, the RTT was about 1ms. The calculated RTT for the test-setup is

[1] A RTT is the time that elapses from when a message is sent from a transmitter to when a response is received back from the receiver[8].

presented in Appendix B. For a real-world setup, a Round Trip Time at about 2ms[2], will have to be added to the measured handover time.

The time consumed for detecting the link breakage will depend on the middleware used for this purpose, and is kept out of the test-setup. Hence, this time will have to be added to the total handover time as well.

The handover time was measured at slightly different CPU speed on the server, but with only the call being measured in progress. The difference in handover time for high and low CPU speed is shown in tables B.1 and B.3 respectively.

There were done 10 test for both high and low CPU speed, and the arithmetic mean and standard deviation was calculated for both scenarios. The arithmetic mean for the handover-time is 42,002ms and 47,809ms for high(2.80GHz) and low(2.10GHz) CPU speed respectively. Based on these measurements, The standard deviation for high and low CPU speed is 3,699ms and 8,053ms respectively. I can not draw any conclusion whether the CPU speed influences the handover time to a substantial extent, due to the high standard deviation and the small number of tests which was performed.

It is likely that the number of UDP sockets contribute to a greater extent to the variation of the handover-time, than the load on the CPU. Due to the lack of equipment, no measurement was done for handover time with a high number of other calls in progress. This should be done to verify that the handover time is not significantly increased with larger load on the server.

Details about the time-measuring is found in Appendix B.

We have chosen to base the solution on a modified client and a corresponding Back to Back User Agent (B2BUA), due to the wish for making a local solution. The advantage of making a local solution, is that we will be able to implement the solution in a controlled environment. If a SIP-proxy was used over the B2BUA, a modification of all-SIP clients which was to be communicated with would have been necessary.

This proposal include a B2BUA like several other SIP-mobility proposals, as described in chapter 3, but our solution is less complex and thus easier to implement. Unlike most other proposals, our solution does not require any additional entities or functionality in the visited domains, nor do we introduce any packet buffering. For that reason, the solution is well suited for real-time applications like voice calls.

[2]RTT measured from UniK to the Norwegian ITSP IP24.

For voice calls, this solution does not put an unacceptable high load on the SIP server, but for the matter of video sessions, the B2BUA solution consumes a not insignificant amount of computation power from the server, and does not scale very well. For scenarios comprising video, especially video transcoding[3], a peer to peer solution would have been a better choice. Despite of the poor scaling properties in a B2BUA, the arguments for a local solution was considered more important.

[3]Video transcoding means to decode the video, and re-encode it using another codec.

Chapter 7

Conclusion

This thesis has described different methods to achieve session continuity in heterogeneous networks, and also proposed an application-layer handover scheme for heterogeneous networks based on the Session Initiation Protocol (SIP). The proposed solution can provide a very low packet loss and handover delay, and with small modifications, a totally seamless handover can be achieved.

Based on the proposed solution, a test-setup was made, where the solution was implemented. Testing and measurements show that the handover-time can be significantly reduced with the proposed solution. From the equation:

$$T_{SIP-Handover} = T_{BreakageDetection} + T_{RetrieveNewIP} + T_{REGISTER} + T_{INVITE}$$

$T_{RetrieveNewIP} + T_{REGISTER}$ will be eliminated, since these actions are already carried out during the set-up of the backup session. T_{INVITE} will be reduced since the decision about media codecs and other parameters is already done. The proposed solution can provide a handover in about 50ms plus the time consumed for detecting the link breakage.

Assumed that the detection of a link-breakage can be done fast enough, the handover delay in the proposed solution is within the limit for the handover to be called "seamless", which is stated to be 200ms.

This solution also has the ability to maintain a session during a link breakage.

This solution is well suited for Fixed Mobile Convergence (FMC) and mobile Voice over IP (VoIP) implementations. It will also be suited for SIP-applications on laptop computers, where the ability to continue the session without interruption when changing from Ethernet to WLAN will be useful.

7.1 Further research

For this solution to work as a practical service, there are several topics that has to be addressed.

To make use of the possibilities of making a perfect seamless handover with the proposed solution, packet filtering and -duplication has to be handled. In my opinion, this should be java-based as well, and integrated with the User Agent. Alternatively, the Linux-software *netfilter*[44], commonly known as *iptables*, can be used. This solution will be significantly easier to implement, but less flexible since it cannot be integrated with the UA, and only work on Linux/UNIX-based operating systems.

Security for VoIP and SIP in particular, is not deeply covered by this thesis. However, support for *Secure RTP* (SRTP) is planned to be included in SIP-Communicator within a short period of time. This will give an significant improvement to the security in the solution.

For the Proactive Handover Solution to work in practice, some kind of middleware which can measure link quality and report to the User Agent is essential. In addition, the algorithm which decides to initiate the handover should be designed in such way that a hysteresis is avoided. Hysteresis might occur when the terminal is traveling through several small zones, while still covered by one large zone. If the handover algorithm is not designed in a way that counter this, the terminal might initiate a large number of unnecessary handovers.

Even though the handover time is measured with load on the B2BUA server, only one call was in progress. The experiments should be repeated with a high number of simultaneous calls to investigate how a large number of UDP sockets contribute to the variation of the handover-time.

Firewall- and NAT traversal have been stated to be out of scope for this thesis, but will eventually have to be considered. Over the past years, several solutions have been proposed to address this problem. IPv6[45] is supposed to eliminate the need for NAT, but no one expects a widespread use of IPv6 in the nearest future. The use of STUN [46] is one way to make SIP work in a NAT-ed network.

Bibliography

[1] 3GPP. *TS 43.318. 3rd Genration Partnership Project; Technical Specifiacation Group GSM/EDGE Radio Access Network; Generic access to the A/Gb interface; Stage 2 (Release 6)*, 6 2006.

[2] Elin Sundby Boysen and Håkon Eyde Kjuus. Proactive Handover in Heterogeneous Networks using SIP. 2007.

[3] Jochen Schiller. *Mobile Communications*. Addison Wesley, second edition, 2001.

[4] WIMAX Forum. Mobile WiMAX Part I: A Technical Overview and Performance Evaluation, 2006.

[5] Muslim Elkotob, Herbert Almus, Sahin Alayrak, and Klaus Rebensburg. The open access network architectural paradigm viewed versus peer approaches. *Teletronikk*, 3_4:33–47, 2006.

[6] Iuliana Popescu. Supporting Multimedia Session Mobility using SIP. In *1st Annual Conference on Communication Networks & Services Research (CNSR2003)*, pages 122–123, Fredericton, New Brunswick, Canada, May 2003. CNSR Project.

[7] F. Chahbour, N. Nouali, and K. Zeraoulia. Fast Handoff for Hierarchical Mobile SIP Networks. *International Journal of Applied Science, Engineering and Technology*, 5:34–37, 2005.

[8] Alberto Leon-Garcia and Indra Widjaja. *Communication Networks*. McGraw-Hill, second edition, 2004.

[9] WINNER II PARTNERS. Winner - wireless world initiative new radio, 2007.

[10] Ed. J. Manner and Ed. M. Kojo. Mobility Related Terminology, June 2004.

[11] 3GPP. *3GPP TS 23.228, IP Multimedia Subsystem (IMS); Stage 2*, 3 2007.

[12] 3GPP. *TS 24.229. 3rd Generation Partnership Project; Technical Specifiacation Group Core Network and Terminals; IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3 (Release 7)*, 6 2006.

[13] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications, January 1996.

[14] International Telecommunication Union. Pulse code modulation (PCM) of voice frequencies, 1988.

[15] International Telecommunication Union. G.722.2 : Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB), 2003.

[16] International Telecommunication Union. Transmission systems and media, digital systems and networks, 1988.

[17] Post og teletilsynet. Det norske ekommarkedet 2006, April 2007.

[18] Tore Riksaasen. *Telematikknett*. Universitetsforlaget, first edition, 1995.

[19] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol, June 2002.

[20] M. Handley and V. Jacobson. SDP: Session Description Protocol, April 1998.

[21] International Telecommunication Union. H.323 Packet-based multimedia communications systems, June 2006.

[22] Mark Spencer et. al. IAX2: Inter-Asterisk eXchange Version 2, October 2006.

[23] Mark Spencer. Asterisk PBX, 2006.

[24] P. Srisuresh and K. Egevang. Traditional IP Network Address Translator (Traditional NAT), January 2001.

[25] umatechnology.org. UMA Participation Companies, 2004.

[26] Norwegian Post and Telecommunications Authority. National table of frequency allocations, 2006.

[27] IEEE. IEEE 802.11 specification, 1999.

[28] Bluetooth SIG. Bluetooth, 2007.

[29] 3GPP. *TS 24.234. 3rd Generation Partnership Project; Technical Specifiacation Group Core Nnetwork and Terminals; 3GPP system to Wireless Local Area Network (WLAN) interworking; WLAN User Equipment (WLAN UE) to network protocols; Stage 3 (Release 7)*, 3 2007.

[30] Network Working Group IETF. Security architecture for the internet protocol, December 2005.

[31] P. Mohana Shankar. *Introduction to Wireless Systems*. Wiley, 2002.

[32] Hello AS. Hello OnePhone, 2007.

[33] Elin Wedlund and Henning Schulzrinne. Mobility support using SIP. In *WOWMOM*, pages 76–82, 1999.

[34] Henning Schulzrinne and Elin Wedlund. Application-layer mobility using sip. *SIGMOBILE Mob. Comput. Commun. Rev.*, 4(3):47–57, 2000.

[35] Nilanjan Banerjee, Sajal K. Das, and Arup Acharya. SIP-Based Mobility Architecture for Next Generation Wireless Networks. In *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pages 181–190, Washington, DC, USA, 2005. IEEE Computer Society.

[36] Paolo Bellavista, Antonio Corradi, and Luca Foschini. SIP-based Proactive Handoff Management for Session Continuity in the Wireless Internet. *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops*, 2006.

[37] Ed C. Perkins. IP Mobility Support for IPv4, August 2002.

[38] Emil Ivov. SIP-Communicator, 2007.

[39] CounterPath. X-lite 3.0 FREE Softphone, 2007.

[40] The Eclipse Foundation. Eclipse IDE, 2007.

[41] NIST M. Ranganathan. jain-sip: JAVA API for SIP Signaling, 2007.

[42] Inc. Sun Microsystems. Java Media Framework API (JMF), 2007.

[43] Gerald Combs. Wireshark, 2006.

[44] Harald Welte (head), Jozsef Kadlecsik, Martin Josefsson, Patrick McHardy, Yasuyuki Kozakai, and Pablo Neira Ayuso. The netfilter.org project, 2007.

[45] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification, December 1998.

[46] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs), March 2003.

Appendix A

Source code

SIP-communicator has a highly extensive source code, and for that reason the code is not presented here. The complete source code is available on the enclosed CD-ROM. On the CD is also the modified code and compiled version of JAIN-SIP.

Appendix B

Measuring

The following equipment is used in the measurement of handover-time in the proposed solution:

B2BUA-server

- Intel(R) Pentium(R) 4 CPU 2.80GHz
- RAM 512MB
- linux-2.6.18-gentoo-r2
- asterisk-1.2.13

Unmodified client

- Intel(R) Core(TM)2 CPU 2.00GHz
- RAM: 1GB
- Windows XP
- X-Lite Version 3.0 build 41150

Modified client

- Intel(R) Core(TM)2 CPU 1.20GHz
- RAM: 1GB
- Windows XP
- SIP-Communicator 1.0-alpha2-0

The handover time was measured on computer running the modified softphone, using the Wireshark Network Protocol Analyzer [43]. In addition, the packet spacing was measured on the B2BUA-server, in order to be able to compute the processing time for the handover in the server.

The test results for the server at high CPU speed (2.8GHz) is shown in table B.1.

Test no.	Time(client)[ms]	Time(server)[ms]	Time Client-Server)[ms]
1	41,728	40,518	1,210
2	37,775	36,253	1,522
3	43,743	40,378	3,365
4	40,993	39,226	1,767
5	45,639	44,367	1,272
6	37,483	36,279	1,204
7	49,230	47,800	1,430
8	43,062	41,428	1,634
9	42,120	40,569	1,551
10	38,243	36,556	1,687
SUM	420,016	403,374	16,642

Table B.1: Results, high CPU speed

Calculation gives the arithmetic mean and the standard deviation for the values. This is shown in table B.2.

Value	Time (client) [ms]	Time (server)	Time (Client - Server)
Arithmetic mean(μ)	42,002	40,337	1,475
Standard deviation(σ)	3,699	3,680	0,209

Table B.2: Calculated values, high speed

The test results for the server at low CPU speed (2.1GHz) is shown in table B.3.

Calculation gives the arithmetic mean and the standard deviation for the values. This is shown in table B.4.

Test no.	Time(client)[ms]	Time(server)[ms]	Time Client-Server)[ms]
1	42,938	41,398	1,540
2	48,588	46,823	1,765
3	37,414	35,922	1,492
4	45,893	44,576	1,317
5	45,454	43,044	2,410
6	43,358	41,560	1,798
7	40,492	38,905	1,587
8	54,718	48,996	5,722
9	64,204	62,802	1,402
10	55,033	53,666	1,367
SUM	478,092	457,692	20,400

Table B.3: Results, low CPU speed

Value	Time(client)[ms]	Time(server)[ms]	Time(Client-Server)[ms]
Arithmetic mean(μ)	47,809	45,769	2,040
Standard deviation(σ)	8,053	7,844	1,332

Table B.4: Calculated values, low speed

Appendix C

SIP message format

C.1 Format

The SIP messages are either requests from a server or client, or responses to a request. The general message format is:

Start Line
Header1: value1 Header2: value2 Header3: value3 ..
Body(optional)

C.2 Requests

RFC 3261 defines six types of requests:

1. INVITE - Used to initiate or modify sessions
2. ACK - Used to acknowledge a received message
3. BYE - Used to terminate a session
4. CANCEL - Used to cancel the previous request which is not yet acted upon completely
5. OPTIONS - Used as a query of options and capabilities, and also in conjunction with NAT/firewall issues.

6. REGISTER - Used to register on a registrar server

Other RFC's extends this set of methods to support notification, event changes and instant messaging. These are however, out of scope in this thesis.

C.3 Responses

There are six classes of responses. The first digit of the Status-Code defines the class of response. For example, a response with status code between 200 and 299 is referred to as a "2xx response".

1. SIP 1xx: Provisional - request received, continuing to process the request
2. SIP 2xx: Success - the action was successfully received, understood, and accepted
3. SIP 3xx: Redirection - further action needs to be taken in order to complete the request
4. SIP 4xx: Client Error - the request contains bad syntax or cannot be fulfilled at this server
5. SIP 5xx: Server Error - the server failed to fulfill an apparently valid request
6. SIP 6xx: Global Failure - the request cannot be fulfilled at any server

Appendix D

Wireshark traces

Wireshark trace for measurement of handovertime is shown in figure D.1

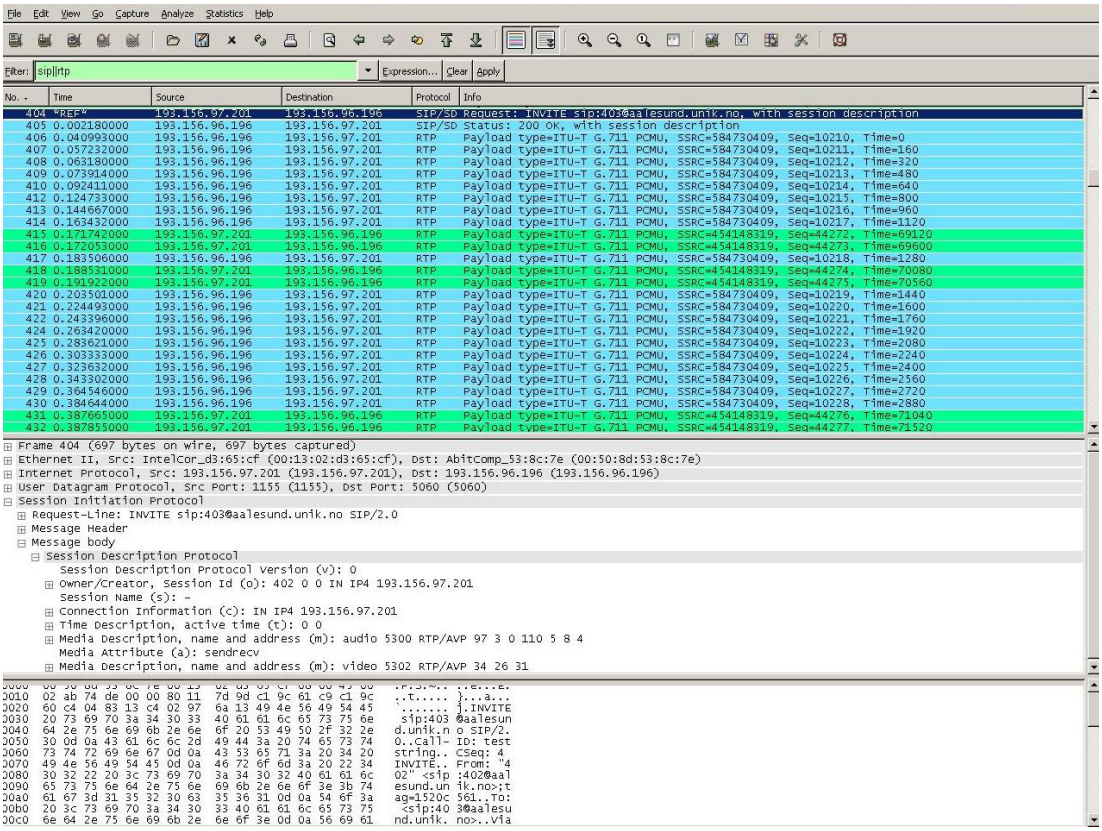

Figure D.1: Wireshark trace