İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

SIP BASED MOBILE VOICE OVER IP CLIENT FOR WIRELESS NETWORKS

M.Sc. Thesis by

Kamil ERMAN, B.Sc.

Department : Computer Engineering

Programme: Computer Engineering

JANUARY 2008

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

SIP BASED MOBILE VOICE OVER IP CLIENT FOR WIRELESS NETWORKS

M.Sc. Thesis by

Kamil ERMAN, B.Sc.

(504041544)

Date of submission : 24 December 2007

Date of defence examination: 29 January 2008

Supervisor (Chairman): Prof. Dr. A. Coşkun SÖNMEZ

Members of the Examining Committee Prof. Dr. Sema Oktuğ

Assist. Prof. Dr. A. Gökhan Yavuz

JANUARY 2008

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ AĞLAR İÇİN SIP TABANLI, MOBİL, IP ÜZERİNDEN SES İSTEMCİSİ

YÜKSEK LİSANS TEZİ

Müh. Kamil ERMAN
(504041544)

Tezin Enstitüye Verildiği Tarih : 24 Aralık 2007
Tezin Savunulduğu Tarih : 29 Ocak 2008

Tez Danışmanı : **Prof. Dr. A. Coşkun SÖNMEZ**
Diğer Jüri Üyeleri **Prof. Dr. Sema Oktuğ**

Assist. Prof. Dr. A. Gökhan Yavuz

OCAK 2008

ACKNOWLEDGEMENT

This thesis is the completion of my MS study at Istanbul Technical University, where I benefit from inspiring atmosphere. I am very grateful to many people who contribute this thesis directly or indirectly. I would like to take this opportunity to thank all of them.

First of all, I would like express my gratitude to my supervisor Prof. Dr, A. Coşkun SÖNMEZ, who gave me the opportunity to do this thesis. He gave me the freedom to try my new ideas about VoIP.

I would also like to thank my colleagues from Nortel Netaş and Telenity. I learned a lot from them about telecommunication issues and applications. They were always there, either to discuss some topics related with my master thesis or just to have an ordinary chat.

My best supporters are the ones I do not see everyday and certainly not at work: my family. I am very thankful to my parents, Zühra and Eyüp ERMAN for their infinite support, and thrust during my whole life. I, also, wish to thank to my sisters, Kezban BAKIRCI and Fatma EROĞLU, who have encouraged me to start my master study after graduation from Yeditepe University, for their unconditional support.

Last but not least, I would like to express my deepest gratitude to my wife, Ayşegül Tüysüz Erman, for her love, unconditional compassion and fully support.

December 2007 Kamil ERMAN

CONTENTS

LIST OF ABBREVIATIONS

2G : Second Generation
3G : Third Generation
3GPP : 3rd Generation Partnership Project
AIN : Advanced Intelligent Network
DAO : Database Access Objects
DPMDN : Digital Public Mobile Data Network
GPRS : General Packet Radio Service
CSCF : Call State Control Function
CSM : Call Setup Manager
GSM : Global System for Mobile Communications
HSS : Home Subscriber Server
HTTP : Hyper Text Transfer Protocol
IETF : Internet Engineering Task Force
IM : Instant Messaging
IMS : IP Multimedia Subsystem
IP : Internet Protocol
ISDN : Integrated Services Digital Network
LDAP : Lightweight Directory Access Protocol
MGCP : Media Gateway Control Protocol
OMA : Open Mobile Alliance
OS : Operating System
PoC : Push to talk over cellular networks
PTT : Push to talk
QoS : Quality of Service
RADIUS : Remote Authentication Dial-In User Service
RSVP : Resource Reservation Setup Protocol
RTP : Real-time Transport Protocol
SDP : Session Description Protocol
SHM : Seamless Handover Mechanism
SIP : Session Initiation Protocol
SS7 : Signalling System #7
TRIP : Telephony Routing over Internet Protocol
UA : User Agent
UMTS : Universal Mobile Telecommunication System
URI : Uniform Resource Identifier
UTRAN : UMTS Terrestrial Radio Access Network
VoIP : Voice over Internet Protocol
VoWiFi : Voice over Wireless Fidelity
Wi-Fi : Wireless Fidelity
WLAN : Wireless Local Area Network

LIST OF TABLES

LIST OF FIGURES

KABLOSUZ AĞLAR İÇİN SIP TABANLI, MOBİL, IP ÜZERİNDEN SES İSTEMCİ

ÖZET

Çeşitli ikinci nesil (2G) ve üçüncü nesil (3G) hücresel ağ teknolojileri ile kablosuz yerel alan ağı (WLAN) ve kablosuz bağlantı (Wi-Fi) teknolojilerinin birlikte işlerliği yaygınlaştığı ve popülarite kazandığı için, çoklu ortam (multimedya) uygulamaları kullanıcıya hissettirmeden farklı hava arayüzleri arasında çalışabilme özelliğine sahip olmalıdırlar. Her iki teknolojinin tek bir istemcide desteklenmesi, bu iki teknolojinin yararlarını birleştirir. Bu çoktürel ortamdan yararlanacak anahtar teknolojilerden biri de IP üzerinden ses (VoIP) iletimi teknolojisidir. Internet Protokolü üzerinden sayısallaştırılmış ses paketinin nasıl taşınacağını tanımlayan VoIP, üçüncü nesil (3G) hücresel teknolojiler ve kablosuz yerel alan ağı (WLAN) gibi belirli bant genişliğine sahip ve kablosuz ortamların diğer kısıtlamalarını içinde taşıyan telsiz teknolojilerden faydalanır.

Hücresel bağlantılılık geniş kapsama alanı sağlarken, düşük bant genişliği ve yüksek gecikmeden yana sıkıntı yaratır. Diğer taraftan, WLAN daha yüksek bant genişliği ve daha düşük gecikme sunar. Bundan dolayı, hücresel ağlarda çalışan VoIP uygulamaları bant genişliğini etkin olarak kullanabilmek için yarı-çift yönlü iletişim tabanlı tasarlanırken, kablosuz yerel alan ağında işleyen VoIP uygulamaları ses kalitesini arttırmak için tam-çift yönlü iletişim kullanırlar. Yarı-çift yönlü VoIP iletişimi temel olarak, diğerleri dinlerken sadece bir kullanıcının konuşabildiği "hücresel ağlar üzerinden bas-konuş" (PoC) yöntemidir. Yarı-çift yönlü iletim yapan IP tabanlı ağ kullanımı daha az bant genişliği gereksinimi yararına sahiptir (örneğin; ağ ve hizmet tasarrufu).

Bu tez SIP tabanlı mobile bir VoIP istemcisinin tasarımını ve gerçeklenmesini tanımlar. Bu tez temelde çoktürel ağlar üzerinde çalışabilen bir VoIP istemcisi tasarımının çözülmesi gereken iki sorununun üzerinde yogunlaşır. Birinci ve en zorlu sorun farklı erişim teknolojileri arasında kullanıcıya fark ettirmeden yer değişim desteği sağlanmasıdır. Bu tezde, kullanıcıya fark ettirmeden el değiştirme yönetimi, uygulama katmanında, multimedya oturumunu başlatmak, sonlandırmak ve değiştirmek için kullanılan Oturum Başlatma Protokolü (SIP) kullanılarak ele

alınmıştır. SIP yaygın bir şekilde kabul edilmekte olan bir VoIP standartıdır. Kullanıcıya fark ettirmeden el değiştirmeyi destekleyebilmek için, VoIP istemcisi üzerinde çalışan SIP tabanlı bir bağlantı yöneticisi önerilmiştir. Bağlantı yöneticisi yeni ağlar keşfettiğinde, adaylar listesinden bir ağ seçer ve hali hazırda yürütülmekte olan iletişimi kullanıcıya fark ettirmeden yeni ağ arayüzüne aktarır. Dolayısı ile, bu birim Wi-Fi, 3G gibi çoktürel ağlar arasında dolaşmayı sağlar.

İkinci sorun ise, en kaliteli çağrı (arama) desteğini sağlamaktır. En kaliteli çağrı desteği, iletişim kurmak isteyen tarafların farklı türden ağlara bağlı olmaları durumunda, VoIP uygulamasının iletişim tipine (yarı-çift yönlü yada tam-çift yönlü) karar verebilmesidir. Örneğin, eğer iletişim kurmak isteyen taraflardan biri bir GSM ağındaysa, en iyi çağrı kalitesini yakalayabilmek için, iletişim yarı-çift yönlü olarak kurulmalıdır. Bu tez, bahsedilen özelliği desteklemek için, istemci tabanlı bir karar mekanizması önerir. Bu karar mekanizması, iletişim kurulmak istenen tarafa, istemcinin içinde bulunduğu ağa göre belirlenmiş iletişim tipini içeren bir davet iletisi gönderir. Diğer istemci bu davet iletisini aldıktan sonra, aynı karar mekanizması, iletişimi “bas-konuş VoIP” yada “tam-çift yönlü VoIP” olarak ayarlar.

SIP BASED MOBILE VOICE OVER IP CLIENT FOR WIRELESS NETWORKS

SUMMARY

As the interoperability between various second-generation (2G)/third-generation (3G) cellular technologies and wireless local area network (WLAN)/wireless fidelity (Wi-Fi) technologies gains widespread popularity, multimedia applications would have to interoperate seamlessly among several air interfaces. The support for both technologies in a single client combines their benefits. One of the key technologies that will benefit from this heterogeneous environment is Voice over IP (VoIP). VoIP, which defines how to carry digitized voice packets over Internet Protocol (IP), utilizes wireless technologies as the data networks such as 3G cellular technologies, and WLAN with the given bandwidth and other constraints of wireless environments.

Whereas cellular connectivity provides wide-area coverage, it suffers from low bandwidth and high delay. On the other hand, WLAN offers higher bandwidths and lower delay. Therefore, while the VoIP applications in cellular networks are based on half-duplex communication to utilize the bandwidth efficiently, the VoIP applications which operate in WLAN employ full-duplex communication to increase the voice quality. The half-duplex VoIP communication is basically the “push to talk over cellular networks” (PoC) where only one user talks while the others listen. Use of the underlying IP-based network with half-duplex transmissions has the advantage of requiring less bandwidth (e.g. network and facility savings).

This thesis describes the design and the implementation of a SIP-based mobile VoIP client. It mainly focuses on two challenges of designing a VoIP client which works on heterogeneous network environments. One and the most challenging problem is the provision of seamless mobility support among different access technologies. In this thesis, seamless handover management is handled at the application layer by using Session initiation protocol (SIP), which is used to initiate, terminate, and modify multimedia session. SIP is becoming a widely accepted standard for VoIP. To support seamless handover, a SIP based connection manager is proposed on VoIP client application. As new networks are discovered by the connection manager, it

selects a new network from the candidate list and transfers the current communication to the new network interface seamlessly. Therefore, this module provides roaming across heterogeneous networks such as Wi-Fi, 3G.

Second problem is providing the best effort call quality support. It means that if the communication parties are in dissimilar networks, the VoIP application should decide the communication type (half-duplex or full-duplex). For instance, if one of the communication parties is in a GSM network, then the communication should be established as a half-duplex manner to achieve best call quality. This thesis proposes a client-based decision mechanism to support this property. This decision mechanism sends an invite message including the communication type (half-duplex or full-duplex) of the client according to the network in which it operates to the other communication party. After the other client receives this invite message, same decision mechanism adjusts the communication as either a “push to talk VoIP” or a “full-duplex VoIP”.

1. INTRODUCTION

The widespread popularity of IP telephony phone calls is increasing day by day, since IP telephony tries to use resources effectively and give much more flexibility for service providers. IP telephony is a combination of hardware and software that enables people to use the Internet as the transmission medium for telephone calls. For users who have free or fixed-price Internet access, an IP telephony software package provides free telephone calls anywhere in the world. IP telephony products can also be called as Voice over IP (VoIP) products.

VoIP uses the Internet to transmit telephone calls by sending voice data in packets using IP rather than by traditional circuit transmissions, called PSTN (Public Switched Telephone Network). The voice traffic is converted into data packets then routed over the Internet, or any IP network as normal data packets would be transmitted. When the data packets reach their destination, they are converted back to voice data again for the recipient.

The VoIP is a key technology that can benefit from the interpretability of WLAN and worldwide deployment of second- and third-generation (2G and 3G) mobile cellular networks. Nowadays, in mobile world, it is a widely used approach, supplying both wireless access interfaces in a single device. Such a device used for VoIP service benefits from higher bandwidth and lower cost of WLAN, and wide-area coverage of cellular connectivity.

In cellular architecture, voice is carried out over assigned channels. For this reason, even if the channel is not used by reserved user, resource is still reserved and unshared for other users. Therefore, conversation quality is quarantined, but resource efficiency is decreased significantly. On the other hand, WLAN cannot provide wide-area coverage. As a result, a user, who wants to make voice communication, would be able to switch between these different networks to carry out best effort voice communication successfully. There are some studies under development which would switch seamlessly from cellular networks to WLAN without dropping a call.

The most challenging problem for this system integration is the provision of seamless mobility support among different access technologies. Several protocols have been proposed for handling wireless mobility; however, usually two basic approaches are considered for VoIP services: Mobile IP and Session Initiation Protocol (SIP) which handle mobility at the network layer and at the application layer, respectively.

Since Third Generation Partnership Project (3GPP) adapts SIP as a call control protocol to manage real-time multimedia sessions within the Internet Multimedia Subsystem (IMS) that is a new framework for providing IP multimedia services, SIP have gained more importance and attention.

This thesis presents a VoIP client which handles handover between Wi-Fi/WLAN and cellular networks (i.e. GPRS, UMTS) in a wireless overlay scenario. Mobility is handled at application layer by using SIP. Also, the disadvantages of Mobile IP for mobility handling are discussed in the next section. Moreover, our VoIP client can choose different conversation types (half-duplex or full-duplex) according to its access technology. In addition, it is possible to change the current conversation type of the communication when the user changes it access network during an active call. For this purpose, we, also, propose a new push-to-talk (half-duplex) communication model.

The rest of this thesis is organized as follows. The rest of this chapter gives an overview about the related works and the approach of the thesis. Chapter 2 summarizes background information about protocols, development environments like operating system and platform, and the programming language which are used in the implementation of proposed VoIP client. Chapter 3 gives the details of the architecture of proposed SIP-based mobile VoIP client. It explains all the components of the model and discusses the differences from existing approaches. Chapter 4 describes in detail the implementation of our mobile VoIP client. Finally, Chapter 5 concludes the thesis and discusses future works.

1.1. Current Studies and Applications

In this section, we mention about some commercial and academic studies for different VoIP applications and Push-to-Talk (PTT) services. Firstly, current VoIP applications supporting mobility among different wireless network interfaces is discussed. One of the most important problems for mobile VoIP clients is providing shortest and successful handover between different networks.

Mobile IP (MIP) [1] is a well known solution for handover management. It is a standard proposed by the Internet Engineering Task Force (IETF). It enables a TCP connection to remain alive and receive packets when a mobile host moves from one point of attachment to another as illustrated in Figure 1.1. Although MIP is a widely used approach applied to VoIP applications, it has some important drawbacks including high handover latency, high packet lost rate, and inefficient routing path.

The most important disadvantage of MIP for real time communication is high handover delay. This well-known problem with MIP is due to the triangular routing which adds delay to the traffic towards the mobile host, but not from the mobile host. Moreover, if either a home agent or a foreign agent becomes a bottleneck due to large number of mobile nodes, delays will be increased due to buffering. Therefore, MIP is an unsuitable protocol for real time communication.

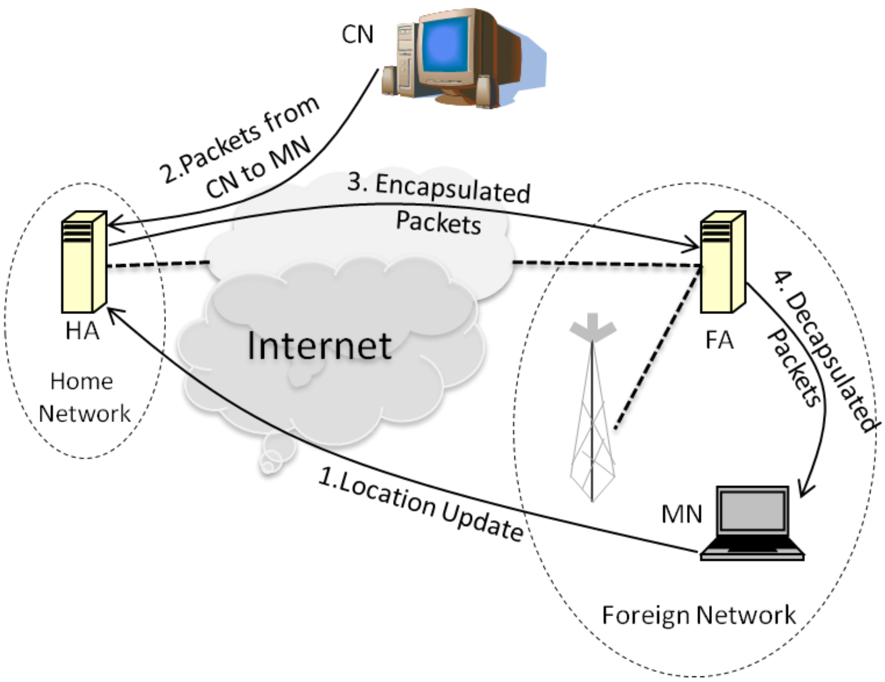

Figure 1.1: Mobile IP routing path

Another protocol proposed to achieve seamless handover is Session Initiation Protocol (SIP). Indeed, SIP based handover proposals are more suitable handover mechanisms for mobility support, since SIP supports personal mobility, i.e., a user can be found independent of location and network device (PC, laptop, IP phone, etc.). The step from personal mobility to IP mobility support is basically the roaming frequency, and that a user can change location (IP address) during a traffic flow.

In [2], Wedlund et al. propose to use application layer protocol SIP in order to support mobility and real-time communication in a more efficient way. When the mobile client changes the location, it sends registration message to the SIP server. When the corresponded node sends INVITE message to the SIP server, SIP server redirects INVITE message to the new location of the mobile client.

In [3], the author mentions about SIP mobility support for a scenario in which a user agent can hand off a call between, for example, a commercial wireless network and a home or office 802.11 wireless network. This scenario requires the actual route set (set of SIP proxies that the SIP messages must traverse) to be changed; therefore, a re-INVITE cannot be used. It needs more than just changing the Contact URI, but needs creation of a new dialog. The solution to this is to send a new INVITE (which creates a new dialog and a new route set including the new set of proxies) with a "*Replaces header*", which identifies the existing session.

The work of Pangalos et al. [4] describes and tests an end-to-end SIP based approach employed when a vertical handover is occurred. The authors propose two adaptation schemes: (i) a receiver based adaptation, and (ii) a proxy based adaptation. The receiver based proposal is to have the receiver part of the communication adapt according to its current network connection. For example, if a movement to a GSM network occurs, then the receiver application would remove the video data stream and compress the audio stream. The proxy based approach employs M3A approach that is an extended version of Mobile IP with smart routing and proxy filtering decisions being taken at the Home Agent (HA). The authors also present their evaluation results in terms of connection delay to new network, home agent registration delay, and overloading of the networks.

Jung et al. [5] present simulation results of MIP and SIP from the perspective of VoIP service in wireless Internet access, and also propose an integrated model that reduces the handover latency and packet loss during handover. The proposed model called SIP-MIP uses a SIP network server, and a Mobility Agent with SIP Registrar to facilitate location management. While the SIP network server handles call/session delivery, the Mobility Agent with SIP Registrar is used for handling location registration, location updates, and location queries. Basically, when the client gets a new mobile IP address, it sends this new IP to the SIP server and then new re-INVITE message is sent to corresponding node.

Dutta et al. introduce an application layer technique to achieve fast-handoff for real-time RTP/UDP-based multimedia traffic in a SIP signaling environment [6]. This technique is based on standard SIP components such as user agent and proxy which usually participate to set up and tear down the multimedia sessions between the mobiles. Unlike network layer techniques, this application layer technique does not have to depend upon any additional components such as home agent and foreign agent. Therefore, it provides a network access independent solution suitable for application service providers.

Sattari et al. [7] compares the handover performance of SIP, MIP, and hybrid models in similar situations (i.e. between WLAN and UMTS networks). The authors also propose a network architecture in which users will be able to choose a combination of several networks to receive a particular service and preserve their session under various mobile conditions. The architecture basically proposes to separate non-real time traffic and real time traffic. If the traffic is non-real time, MIP is used. If the traffic is real time, SIP handover mechanism is used.

Rajkumar et al. discuss a network architecture that supports seamless interoperability between 3G and WLAN and propose autonomous layer two enhancements at the client side to support SIP-based multimedia applications in general and voice applications in particular [8]. The proposal is for an intelligent agent along with an mobile client that can simultaneously monitor wireless signal strengths of multiple access technologies and automatically perform access technology handovers. Indeed, the paper proposes a handover decision mechanism and discusses the essential elements of a system that would deliver VoIP in heterogeneous networks. It does not propose communication architecture for VoIP applications.

The study of Bellavista et al. [9] describes how to exploit the SIP to build a middleware for session continuity during handover. This proposed middleware makes use of SIP notifications to update session information and to proactively activate session reconfiguration for roaming clients.

In [10], Suriol et al. analyze the handover performance of VoIP sessions in a wireless overlay of 802.11 WLANs and GPRS/UMTS networks when mobility is handled at the application layer by SIP. In addition, the authors achieve an acceptable quality for handovers from WLAN towards UMTS. On the other hand, unacceptable values are obtained when moving towards GPRS. The main reason of these unacceptable values is the delay experienced by SIP messages when traversing the cellular network. Their analysis of the different components of the handover delay shows that the cellular network performance is the main factor contributing to the delay when handing over from WLAN to GPRS or UMTS.

Choong et al. present the design, implementation, and deployment issues/options of a handover mechanism using the Intel framework which includes a connection manager and a mixed network IEEE802.21 adaptation layer together with BT's (British Telecom) SIP-based audio/video application in a heterogeneous Wi-Fi, WiMAX and Ethernet network environment [11].

Secondly, we want to discuss some studies about push-to-talk. Push to talk over cellular (PoC) likes a walkie-talkie that is provided over a cellular phone network. There are many academical and commercial researches about PoC. First commercial research, which was performed by Nextel Communications and Motorola phones, was used as the first PoC client. This service was operated in a region having a radius of up to six miles. In 2000s, this service was so popular and Mobile Tornado, Motorola, Nokia, Ericsson, Siemens, Sonim and Wireless ZT published their own PoC solution. These solutions were based on 2G and 3G networks and used SIP and RTP protocols. To provide interoperability, Open Mobile Alliance defined a PoC service as a part of the IP Multimedia Subsystem (IMS), and the first OMA (Open Mobile Alliance) standard about PoC was finalized in first half of 2005. On the other hand, there is no 100% compatibility list available for PoC services, because Motorola, Nokia, Ericsson, Siemens AG and AT&T Mobility have their own PoC solutions which are still used by some GSM operators.

In the literature, researches are focused on interoperability of Push to talk (PTT) solutions and propose new SIP based solutions for PoC. In [12], Parthasarathy presents implementation of a Java based PoC solution for 2.5G networks. The author defines call-initiation, call-termination and registration mechanism for one-to-one server or client communication. An architecture called 3PoC is proposed and evaluated in [13]. 3PoC is 3GPP based system architecture for efficient implementation of PoC services in 3G packet switched networks. The paper also investigates impacts of PoC requirements on 3GPP UTRAN (UMTS Terrestrial Radio Access Network) and CN (Core Network) network elements. In addition, the performance of PoC signaling transfer over the proposed architecture is evaluated by a simulation in the paper. Siemens PoC solution presented in [14] shows the design of a Push-to-Talk service operated over a GPRS/UMTS network and measurement results achieved in a live network based on existing GPRS access technology. Figure 1.2 redrawn from [14] shows an example service operation for PoC Instant Personal Talk. It illustrates SIP and RTCP Floor Control signalling between PoC Servers in more detail.

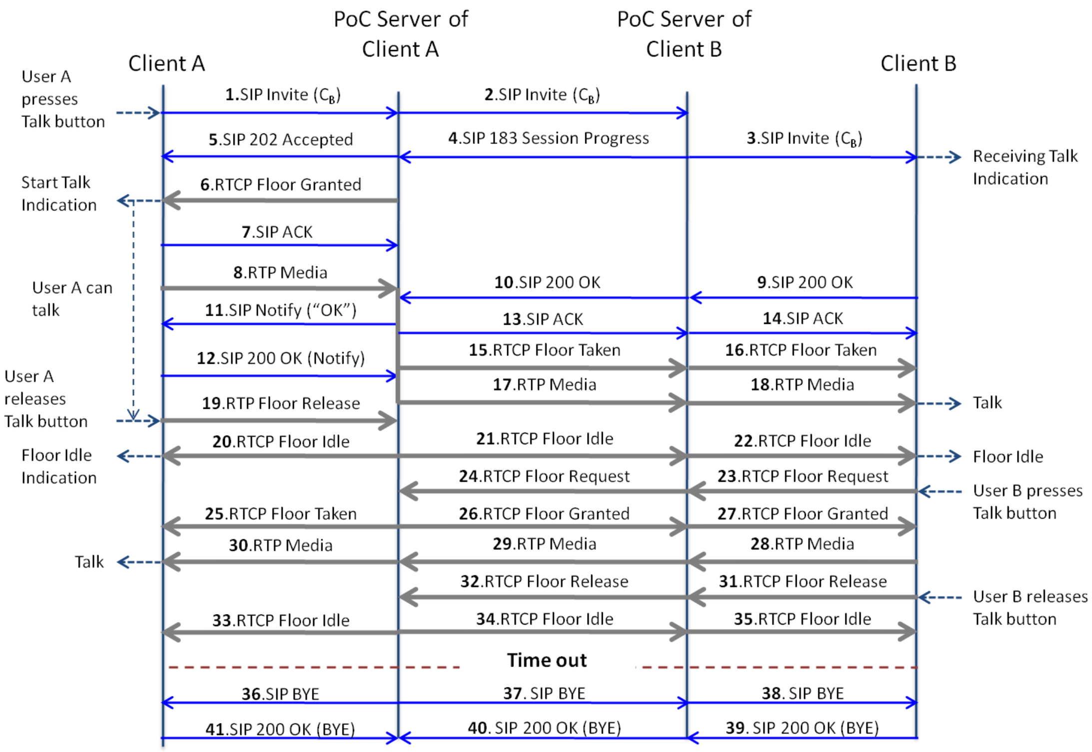

Figure 1.2: PoC Session Setup for an Instant Personal Talk and Auto-Answer Procedure

In [15], Peng et al. evaluate the structure and performance of the PTT service using SIP as call control protocol for 2.5G DPMDN (Digital Public Mobile Data Network) such as GPRS and cdma2000 1X , and present their improved call control protocol. Their new call control protocol, which is based on the SIP-based PTT shown in Figure 1.3, tries to decrease the call delay time which is one of the most key factors that reuse the channel again. The proposed model makes use of the user's pre-knowledge of PTT talking model and applies a one-way signaling structure by elimination two-way acknowledge procedure in SIP-based scheme illustrated in Figure 1.3. In contrast to messaging in the figure, because this model uses the pre-knowledge of PTT call model, even when user A and user B press the button and speak almost at the "same" time, the server will give the first reach one (for example user A) the call right and discard the other (for example user B).

Blum et al. consider other media types than voice, like video communication, file transfer or service subscription for content push services, which are named as Push-to-MultiMedia (PTM), and add them as PTT services [16]. The study reports a concept of integrating PTT/PTM functionality in community-based applications to enable already existing groups and communities with new communication features. In [17], the authors describes an IMS social network application that can be used to interact, communicate, inform themselves and others about news, events etc. within their community using IMS enabled cell phones. The proposal implements a new J2ME compatible PTT instead of using existing Siemens PoC. The implementation, which does not include any new PTT features, covers very simple PTT services considering only audio data transmission.

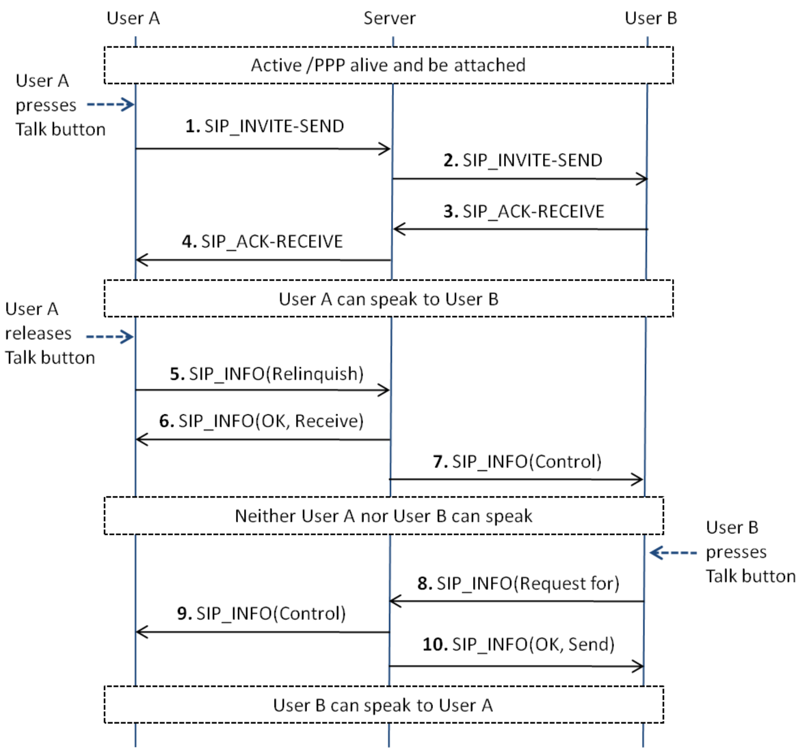

Figure 1.3: One-to-one push to talk

1.2. The Approach

This thesis proposes a mobile VoIP client for heterogeneous wireless networks. The objective of this thesis is to describe the problems of existing VoIP applications and to design and implement a new mobile VoIP client.

The main contributions of this thesis can be listed as follows:

- Discussing different types of VoIP approaches and their drawbacks.
- Designing and implementing a new SIP-based handover management model to support mobility among different access technologies. The aim is to achieve seamless handover between different networks (e.g. 3G to WLAN). Here by seamless, we mean low latency and low packet lost during handover process. A SIP-based connection manager is proposed on VoIP client application for this purpose.
- Designing and implementing a new client-based decision mechanism to support best call quality. It basically decides the communication type of VoIP (i.e. half-duplex or full-duplex) according to the access networks of communication parties.
- Designing and implementing a new SIP-specific event-based push-to-talk service.

In this thesis, to achieve second and third goals, and to build a complete multimedia architecture and develop a mobile VoIP client for wireless networks, Session Initiation Protocol is used with other IETF protocols such as the Real-time Transport Protocol for transporting real-time data and, the Real-Time streaming protocol for controlling delivery of streaming media, and the Session Description Protocol for describing multimedia sessions. Therefore, SIP will be used in conjunction with these protocols (RTP, RTCP, and SDP) in order to provide complete services to the end-users. However, the basic functionality and operation of SIP does not depend on any of these protocols.

Moreover, our final goal points out a new PTT mechanism which differs from existing proposals. All the existing PTT studies depend on server type solutions. It means that all media channel sharing and call control operations are managed by a server. Therefore, any client has to be connected to a server to use push to talk service. Indeed, these solutions use SIP for the call control, and RTCP for the media channel sharing. Thus, RTP media session is redirected from server to client (step 17 in Figure 1.2) and vice versa. This increases the delay because RTP session is not directly established between clients and voice packets are transferred via servers.

In this thesis, a new SIP-specific event-based PTT service for both client-based and server-based approaches is proposed. In our proposal, all call control and media channel sharing operations are controlled by SIP protocol. Therefore, in client-based approach, RTP media session is established directly between clients without connecting to any intermediate server. In the server-based approach, servers are used only for media channel sharing, but RTP media session is only between clients.

2. BACKGROUND AND MOTIVATION

As it is mentioned in the first chapter, the goal of this thesis is designing and implementing a mobile VoIP client for wireless networks. Therefore, an investigation of different protocols supporting VoIP is explained in this section.

VoIP is made of two parts: Signalling (call setup) and audio data transport. VoIP applications generally comply with many protocol standards. Firstly, the VoIP signalling function can be performed by using protocols such as Session Initiation Protocol (SIP) [18], H.323 [19], and Media Gateway Control Protocol (MGCP) [20]. After a long discussion about which call setup protocol is best suited to replace Integrated Services Digital Network’s (ISDN) Signalling System #7 (SS7) infrastructure, the industry has now settled on the SIP, an Internet Engineering Task Force (IETF) standard. In this thesis, SIP is used as an application layer control signalling protocol in the multimedia session operations for our VoIP application. Secondly, data transport can be performed by The Real-time Transport Protocol (RTP) [21] which is considered as the most powerful protocol to deliver multimedia packets in a session. In this thesis, RTP is implemented to exchange voice data during conversations.

Moreover, to describe multimedia sessions Session Description Protocol (SDP) is used, which is a format for describing streaming media initialization parameters.

The rest of this section details SIP, SDP, and RTP which are used in the proposed VoIP application. In addition, other alternative protocols are discussed in the corresponding sections.

2.1. Session Initiation Protocol (SIP)

There are many applications of the Internet that require the creation and management of a session, where a session is considered an exchange of data between associations of participants. The SIP is an IETF standard protocol [18] for initiating an interactive user session that involves multimedia elements such as video, voice, chat, gaming, and virtual reality. SIP can establish, modify, and terminate multimedia sessions (conferences) such as Internet telephony calls. SIP can also invite participants to already existing sessions, such as multicast conferences.

SIP establishes sessions much like the Internet from which it was modeled in order to complete networked messages from multiple PCs and phones. In contrast to the International Telephony Union (ITU) SS7 standard used for call setup and management and the ITU H.323 video protocol suite, SIP operates independent of the underlying network transport protocol and is indifferent to media. Instead, it defines how one or more participant's end devices can create, modify and terminate a connection whether the content is voice, video, data or Web-based.

SIP has major advantages over protocols such as the MGCP, which converts PTSN audio signals to IP data packets. Since MGCP is a closed, voice-only standard, it is complex to enhance it with signalling capabilities and sometimes has resulted in corrupted or discarded messages that discourage providers from adding new services. On the other hand, programmers can use SIP to add new bits of information to messages without compromising connections. For example, a SIP service provider could establish an entirely new medium consisting of voice. With MGCP, H.323 or SS7, the provider would have to wait for a new iteration of the protocol to support the new medium. Using SIP, a company with locations on two continents could enable the medium, even though the gateways and devices may not recognize it. Table 2.1 gives a comparison between these protocols.

Table 2.1: Comparison of different protocols supporting VoIP

	SIP	H.323	MGCP
Philosophy	Horizontal	Vertical	Vertical
Complexity	Low	High	High
Scope	Simple	Full	Partial
Scalability	Good	Poor	Moderate
New Service Revenues	Yes	No	No
Internet Fit	Yes	No	No
SS7 Compatibility	Poor	Poor	Good
Cost	Low	High	Moderate

In addition to these characteristics, because SIP message format is similar to the Hyper Text Transfer Protocol (HTTP) message format, developers can more easily and quickly create applications using popular programming languages such as Java. It takes years to deploy call waiting, caller ID and other services by using SS7 and the Advanced Intelligent Network (AIN); on the other hand, premium communications services can be deployed in just months with SIP.

However, SIP is independent of the details of the session. Sessions here mean a set of senders and receivers that communicate and the state they are kept in during the communication. Therefore, SIP isn't all in one approach; it just makes the communication possible, but the communication itself is achieved by other means.

SIP is neither a session description protocol, nor does it provide conference control. The main advantage is that it is compatible with different architectures and deployment scenarios in the Internet services. Its essential communication function is aided by extensions and further protocols and standards. To describe the payload of message content and characteristics, SIP uses the Internet’s SDP to describe the characteristics of the end devices. SIP, also, does not itself provide Quality of Service (QoS) and interoperates with the Resource Reservation Setup Protocol (RSVP) for voice quality. It also works with a number of other protocols, including the Lightweight Directory Access Protocol (LDAP) for location, the Remote Authentication Dial-In User Service (RADIUS) for authentication and RTP for real-time transmissions, among many others. Figure 2.1 shows the relationship among these protocols and SIP.

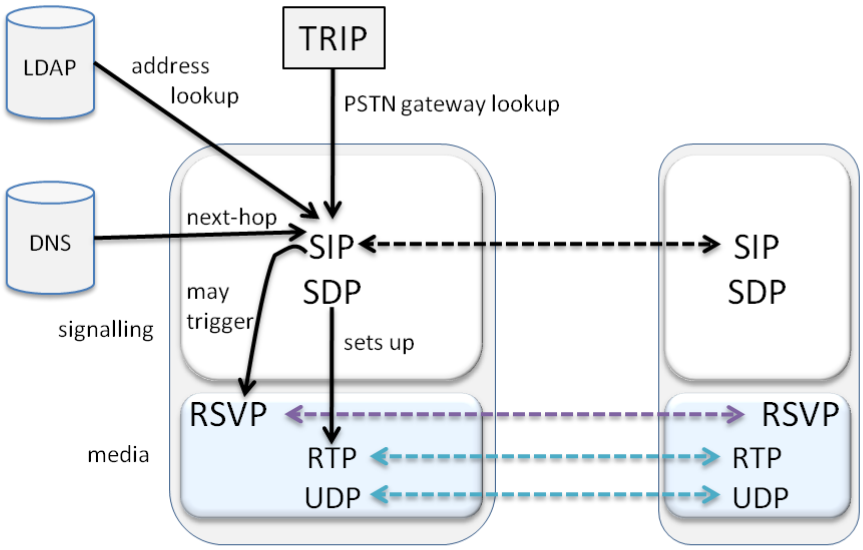

Figure 2.1: Relationship between SIP and other protocols

2.1.1. SIP Architecture

SIP sessions utilize up to four major components: (1) *SIP User Agents*, (2) *SIP Registrar Servers*, (3) *SIP Proxy Servers* and (4) *SIP Redirect Servers*. These protocol components together deliver messages embedded with the SDP protocol, defining their content and characteristics, to complete a SIP session. In the following, a high-level description of each SIP component and the role it plays in this process are presented.

1) **SIP User Agents (UAs)** are the end-user devices, such as cell phones, multimedia handsets, PCs, PDAs, etc. which are used to create and manage a SIP session. Therefore, there is the User Agent Client (UAC) initiating the call message and sending SIP requests and then the User Agent Server (UAS) answering the call. It receives and responds to SIP requests and can accept, refuse or redirect the call. The User Agent software switches between the UAC and UAS modes on a message-by-message basis depending on what is going on.
2) **SIP Registrar Servers** accept registration requests. These servers maintain the databases that contain location information of all user agents registered with a particular SIP domain, thereby enabling the users to update their location and policy information. In SIP messaging, these servers retrieve and send participants' IP addresses and other pertinent information to the SIP Proxy Server.
3) **SIP Proxy Servers** are network hosts acting as both clients and servers to other entities. The job is to ensure requests are routed to appropriate entity identified by a SIP Uniform Resource Identifier (URI). It accepts session requests made by a SIP UA and queries the SIP Registrar Server to obtain the recipient UA's addressing information. It then forwards the session invitation directly to the recipient UA if it is located in the same domain or to a Proxy Server if the UA resides in another domain.
4) **SIP Redirect Servers** receive SIP requests and send response to zero or more addresses. The first location to answer takes the call. Redirect servers do not initiate SIP requests or accept SIP calls. It allows SIP Proxy Servers to direct SIP session invitations to external domains. SIP Redirect Servers may reside in the same hardware as SIP Registrar Servers and SIP Proxy Servers.

The Figure 2.2, redrawn from [22], shows briefly how some of these SIP logical entities use messages to interact. In this figure, the message and action sequence for establishing a SIP session in dissimilar domains (a voice call from a PC (soft-phone) to a (hardware) multimedia handset) is illustrated. The following numbered list explains the arrows with number in the figure.

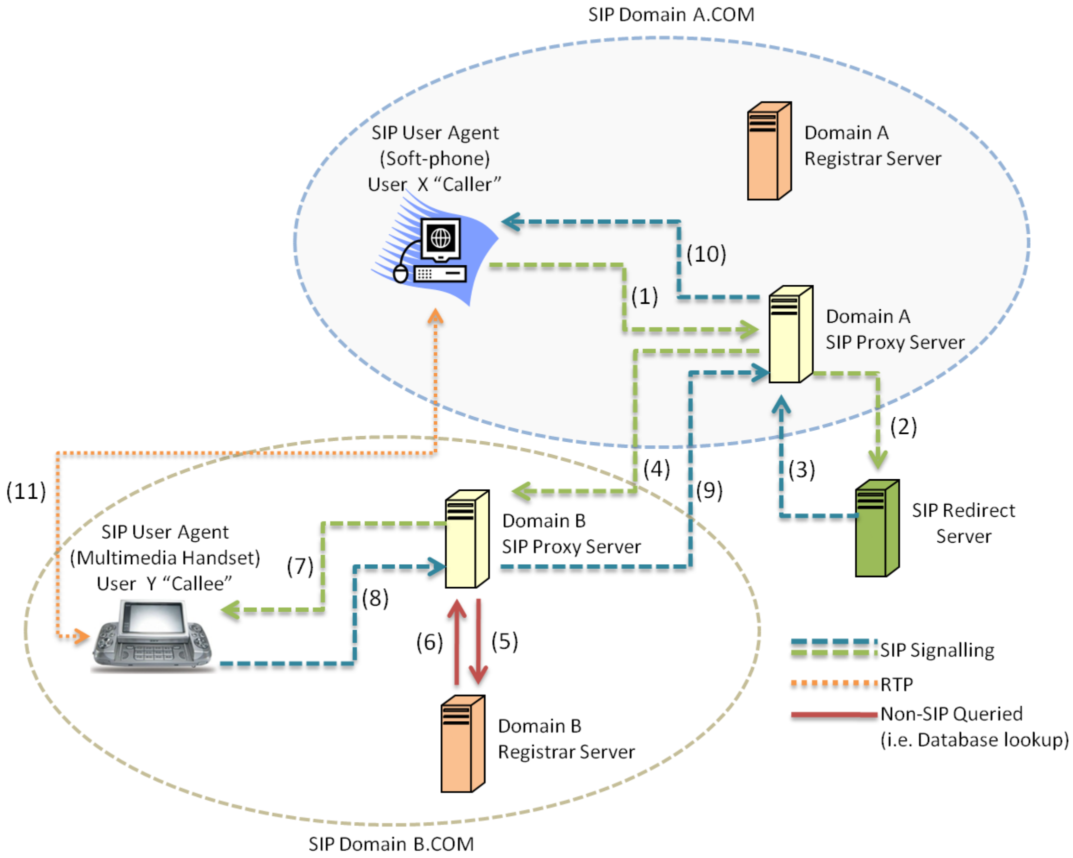

Figure 2.2: Establishing a SIP session in dissimilar domains

1) **Calling:** User X in Domain A calls User Y in Domain B.
2) **Query of Proxy server in Domain A:** The aim of this step to learn how to get to User Y in Domain B. Here, the SIP Proxy Server of Domain A recognizes that the User Y is outside the Domain A and queries the SIP Redirect Server for User Y's IP address.
3) **Redirect Server's response:** IP address of Y is enclosed in the response message coming from Redirect Server to Proxy Server in Domain A.
4) **Forwarding the call to SIP Proxy Server of Domain B:** SIP Proxy Server A forwards the SIP session invitation to SIP Proxy Server B
5) **Query of Proxy server in Domain B:** The aim of this step is to learn where User Y is in the Domain B.

6) **Registrar Server B's response:** The address of User Y is enclosed in this response message of Registrar Server B.
7) **Delivering call request:** Proxy server B delivers User X's invitation to User Y.
8) **Response of User Y:** User Y responds to User X's call.
9) **Response passing I:** Proxy server B of User Y sends the response of User Y to Proxy server A of User X.
10) **Response passing II:** Proxy server A of User X conveys User Y's response to User X. Also, User Y forwards his/her acceptance along the same path the invitation travelled.
11) **Multimedia channel establishment:** If the call set-up is successful (Y is free to accept the call of X), a media path using RPT is established between X and Y and the connected parties can start to talk.

2.1.2. SIP Message Format

SIP is a text based like protocol with a well defined format. This fact facilitates the development of parsing algorithms to interpret messages. SIP Messages are always a Request or a Response. Both message types have a structure divided in three parts: Status Line, Headers and Message Body. The Status Line part differs from Request to Response. The header parts differ slightly from Response to Request while the Body is the same.

The Request Status Line is composed from: Method, Request URI and Version. The Method is one of already defined 6 different commands: INVITE, ACK, OPTIONS, BYE, CANCEL and REGISTER. The Request URI indicates the user to which a request is being addressed. The Version identifies the protocol version number.

The Response Status Line is composed from: Version, Status Code and Reason Phrase. The Version identifies also the protocol version number. The Status Code is a 3 digit number identifying the outcome of the attempt to understand and satisfy a request message. The Reason Phrase is a textual description the response number. There are 6 different defined response classes which are identified by the first digit on the response number. The classes are: 1XX - provisional, 2XX - success, 3XX - forwarding, 4XX - request failure, 5XX - server failure, 6XX - global failure.

The Headers Part is composed from different message fields. Some fields can be used on both Request and Response messages, others only Request messages and others only on Response Messages.

The Message Body is generated using SDP.

The bellow gives a graphic illustration about the message format.

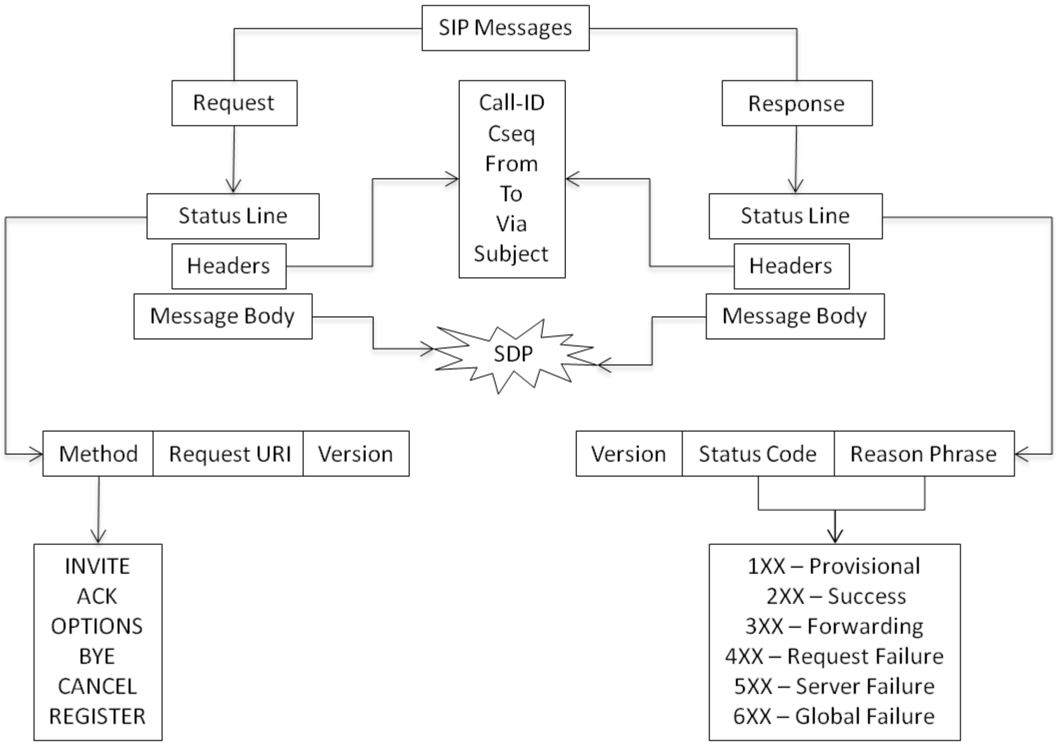

Figure 2.3: SIP Message Format

2.1.3. SIP Mobility

During a session, a mobile device may also change IP address as it switches between one wireless network and another. Basic SIP supports this scenario as well, since a re-INVITE in a dialog can be used to update the Contact URI and change media information in the SDP. This is shown in the call flow of Figure 2.4.

When a proxy server or redirect server receives an INVITE message, it may consult the location server for a route through which to redirect the INVITE message as shown in Figure 2.4. A redirect server provides the calling party with an alternate address for the callee, whereas a proxy server will forward the call to the alternate address on behalf of the caller. A user always belongs to a home network that maintains its home SIP servers. The user re-registers with his/her home network when changing his/her point of attachment. A permanent or persistent IP address is not assumed when the user relocates to a new network. When the mobile user is moving between locations, it is expected that long delays will be introduced while the client is obtaining a new IP address and authenticating with the local authentication, authorization, and accounting servers.

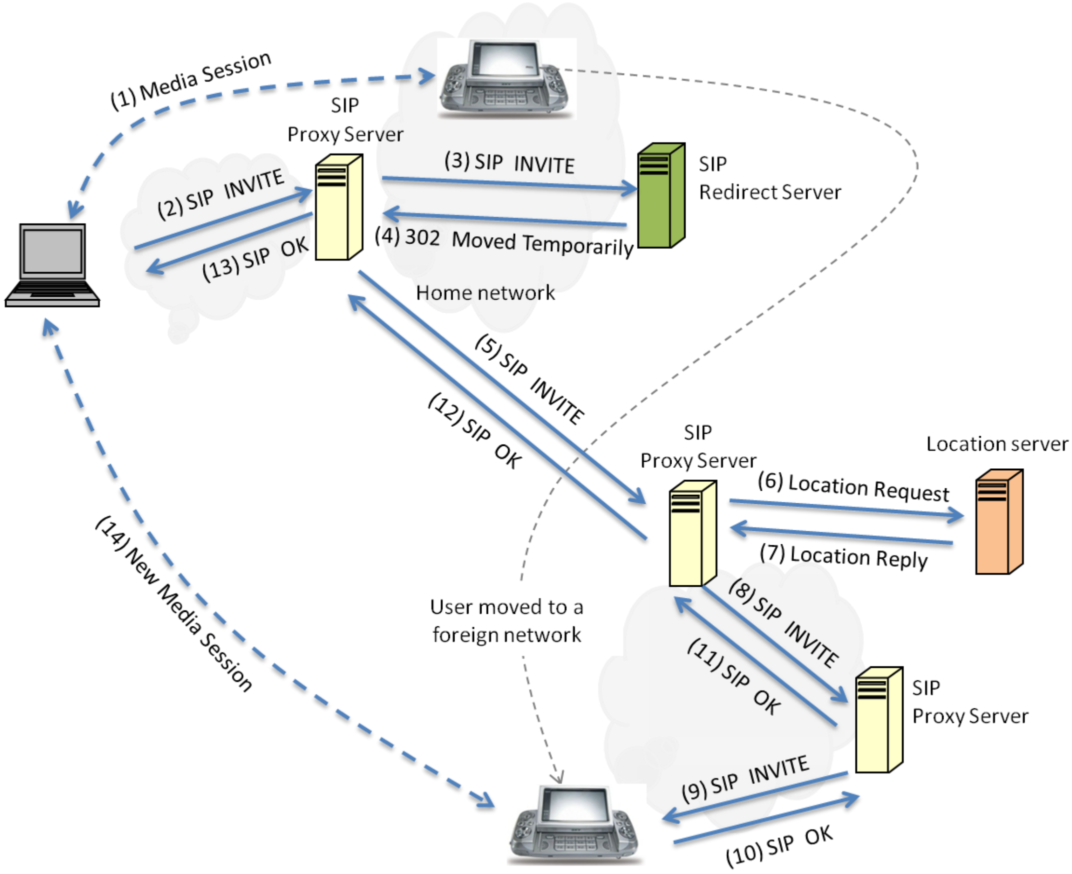

Figure 2.4: Basic SIP mobility using re-INVITE

2.1.4. SIP Characteristics and Functionality

SIP's main function is to help session originators deliver invitations to potential session participants irrespective of where they are. To achieve this SIP uses a wide variety of protocols. Each protocol distinctly addresses the different aspects of the requirement. It also supports user mobility by proxying and redirecting requests to the user's current location. SIP is not tied to any particular conference control protocol. In essence, SIP provides or enables the following functionalities:

Name translation and user location – This ensures that the called party is located and the call reaches it. It carries out any mapping of descriptive information to location information. It ensures that details of the nature of the call (Session) are supported.

Feature negotiation - This allows the group involved in a call (this may be a multi-party call) to agree on the features supported - recognizing that not all the parties can support the same level of features. For example, video may or may not be supported.

Call participant management – This allows the participant to bring other users in to the call or put them on hold or cancel their call while the session is on.

Call feature changes – This enables the user to change call characteristics like *voice-only* facility for one user, *video function* for another during the same session. Also, a third party joining a call may require different features to be enabled in order to participate in the call.

SIP fulfils these functions and re-uses other web elements to make it flexible and scalable.

Rather than defining a new type of addressing system, SIP addresses users by an email-like address. Each user is identified through a hierarchical URL that is built around elements such as a user's phone number or host name (for example, sip:user@company.com). This means that it is just as easy to redirect someone to another phone as it is to redirect someone to a webpage. Figure 2.5 shows the SIP addressing. SIP transparently supports name mapping and redirection services, which supports personal mobility - users can maintain a single externally visible identifier regardless of their network location.

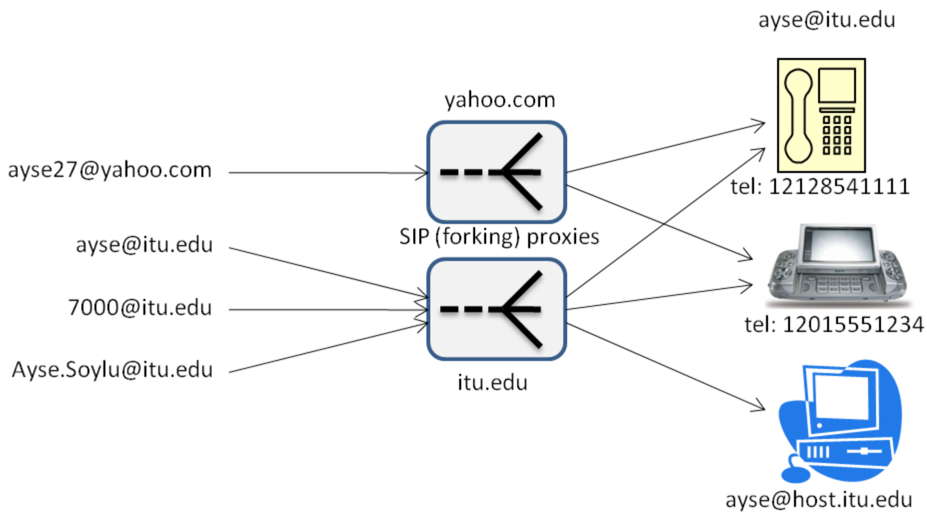

Figure 2.5: SIP email-style addressing

SIP provides its own reliability mechanism and is therefore independent of the packet layer and only requires an unreliable datagram service. SIP is typically used over UDP or TCP.

SIP provides the necessary protocol mechanisms so that end systems and proxy servers can provide services:

- User location
- User capabilities
- User availability
- Call set-up
- Call handling
- Call forwarding, including:
    - The equivalent of 700-, 800- and 900- type calls;
    - Call-forwarding no answer;
    - Call-forwarding busy;
    - Call-forwarding unconditional;
    - Other address-translation services
- Callee and calling "number" delivery, where numbers can be any (preferably unique) naming scheme
- Personal mobility, i.e., the ability to reach a called party under a single, location-independent address even when the user changes terminals
- Terminal-type negotiation and selection: a caller can be given a choice how to reach the party, e.g. via Internet telephony, mobile phone, an answering service, etc.
- Terminal capability negotiation
- Caller and callee authentication
- Blind and supervised call transfer
- Invitations to multicast conference

Developers have also extended the SIP to generate *event notifications* [23]. Users subscribe to an event with the SUBSCRIBE [18] method and receive notifications via NOTIFY messages [18]. Event notification is typically used for presence notification and event signaling during telephone calls. The entities in the network can subscribe to the resource or the call state for various resources or calls in the network, and those entities (or entities acting on their behalf) can send notifications when those states change. The Figure 2.6 shows the typical flow of messages redrawn from [23].

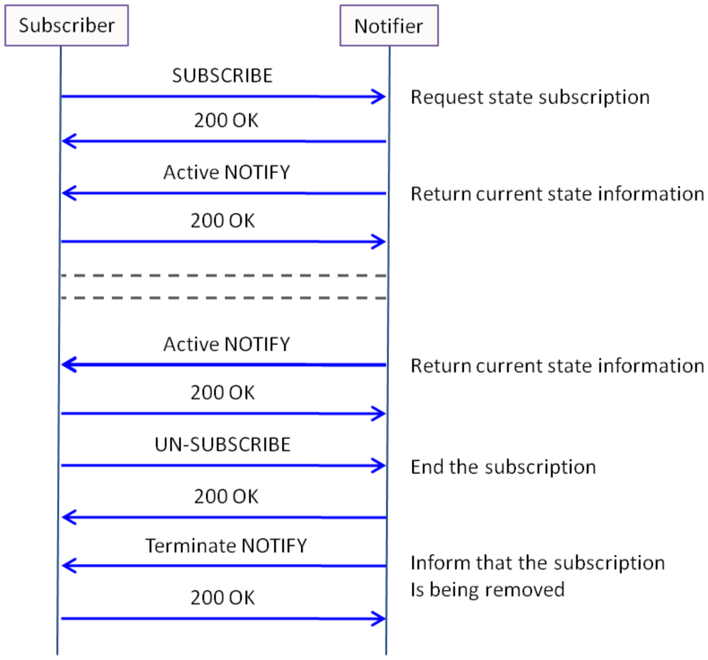

Figure 2.6: Call flow of SIP-Specific Event-Based Notification

2.1.5. SIP Applications

SIP makes possible to connect users across any IP network (WLAN and WAN, the public Internet backbone, mobile 2.5G, 3G and Wi-Fi and any IP device such as phones, PCs, PDAs, mobile handsets). When it is even used alone, SIP-based applications such as VoIP, rich media conferencing, push-to-talk, location-based services are possible to realize as commercial opportunities. However, SIP's ultimate value lies in its ability to combine these capabilities as subsets of larger, seamless communications services.

Using SIP, service providers and their partners can customize and deliver a portfolio of SIP-based services that let subscribers use conferencing, Web controls, Presence, Instant Messaging (IM) and more within a single communications session. Service providers can, in effect, create one flexible application suite that addresses many end user needs instead of installing and supporting discrete, "stovepipe" applications that are tied to narrow, specific functions or types of end devices [24].

- setting up voice-over-IP calls

- setting up multimedia conferences
- event notification (subscribe/notify) →IM and presence
- text and general messaging
- signalling transport

2.2. Session Description Protocol (SDP)

When initiating multimedia teleconferences, voice-over-IP calls, streaming video or other sessions, there is a requirement to convey media details, transport addresses, and other session description metadata to the participants. SDP provides a standard representation for such information, irrespective of how that information is transported. Streaming media is content that is viewed or heard while it is being delivered. SDP, which has been published by the IETF as RFC 4566 [25], is purely a format for session description and initialization of parameters. It does not incorporate a transport protocol, and it is intended to use different transport protocols as appropriate like SIP. For example, SIP describes the communication needed to establish a phone call and the details are then further described in the SDP protocol. This chapter briefly describes the SDP as defined on the RFC 4566.

2.2.1. Requirements

The aim of SDP is to carry information about media streams in multimedia sessions to allow the recipients of a session description to participate in the session. Media streams can be many-to-many. Sessions need not be continually active. An SDP session description includes the following:

- Session name and purpose
- Time(s) the session is active
- The media comprising the session
- Information needed to receive those media (addresses, ports, formats, etc.)

Since resources necessary to participate in a session may be limited, some additional information may also be desirable:

- Information about the bandwidth to be used by the session
- Contact information for the person responsible for the session

2.3. Real-time Transport Protocol (RTP)

The RTP defines a standard packet format for delivering audio and video over the internet. It was developed by the Audio-Video Transport Working Group of the IETF and first published in 1996 as RFC 1889 [26] which was made obsolete in 2003 by RFC 3550 [21].

The RTP provides end-to-end delivery services for data with real-time characteristics, such as interactive audio and video or simulation data, over multicast or unicast network services. It was originally designed as a multicast protocol, but has since been applied in many unicast applications. It is frequently used in streaming media systems (in conjunction with RTSP) as well as videoconferencing and push to talk systems (in conjunction with SIP), making it the technical foundation of the VoIP industry. It goes along with the RTCP and it's built on top of the User Datagram Protocol (UDP). Applications using RTP are less sensitive to packet loss, but typically very sensitive to delays, so UDP is a better choice than TCP for such applications.

RTP consists of two closely-linked parts:

(1) The real-time transport protocol (RTP), to carry data that has real-time properties.

(2) The RTP control protocol (RTCP), to monitor the quality of service and to convey information about the participants in an on-going session. RTCP adds information for packet loss, jitter, delay, signal level, call quality metrics, echo return loss, etc. The latter aspect of RTCP may be sufficient for "loosely controlled" sessions, i.e., where there is no explicit membership control and set-up, but it is not necessarily intended to support all of an application's control communication requirements.

RTP itself does not provide any mechanism to ensure timely delivery or provide other quality-of-service guarantees, but relies on lower-layer services to do so. It does not guarantee delivery or prevent out-of-order delivery, nor does it assume that the underlying network is reliable and delivers packets in sequence. The sequence numbers included in RTP allow the receiver to reconstruct the sender's packet sequence, but sequence numbers might also be used to determine the proper location of a packet.

Table 2.2: RTP packet structure

2	3	4	8	9	16bit	32bit
V	P	X	CSRC count	M	Payload type	Sequence number
Timestamp						
Synchronization source (SSRC)						
Contributing source (CSRC: variable 0 - 15 items, 2 octets each)						

Table 2.2 shows the packet structure of RTP. V (*version*) identifies the RTP version. When P (*padding*) is set, the packet contains one or more additional padding octets at the end which are not part of the payload. When X (*extension*) bit is set, the fixed header is followed by exactly one header extension, with a defined format. *CSRC count* contains the number of CSRC identifiers that follow the fixed header. The interpretation of the M (*marker*) is defined by a profile. It is intended to allow significant events such as frame boundaries to be marked in the packet stream. The *payload type* identifies the format of the RTP payload and determines its interpretation by the application. A profile specifies a default static mapping of payload type codes to payload formats. Additional payload type codes may be defined dynamically through non-RTP means. *Sequence number* increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. *Timestamp* reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations. SSRC (*synchronization source*) is the identifier chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier. CSRC (*contributing source identifiers list*) identifies the contributing sources for the payload contained in this packet.

2.4. Wireless Mobile Networks

The services that are enabled by SIP are equally applicable in the world of mobile networks. A prime example is presence: a user registers their location with a SIP server and the server then knows if the user is available and where the user can be found. Location could be home, work or mobile. The mobile component, then, is crucial if services are to be portable across platforms. As mobile networks evolve, they are becoming more concerned with data and are consequently increasingly IP-centric. With this use of IP comes SIP.

Therefore, this section will give an introduction to some wireless mobile networks which can make use of SIP applications.

2.4.1. Wi-Fi

Wi-Fi is the short form of ***wi**reless **fi**delity* and refers to any system that uses any type of 802.11 network, whether 802.11b, 802.11a, dual-band, etc. In a Wi-Fi network, computers with Wi-Fi network cards connect wirelessly to a wireless router. The router is connected to the Internet by means of a modem, typically a cable or DSL modem. Any user within 61 meters or so of the access point can connect to the Internet, though for good transfer rates, distances of 30.5 meters or less are more common.

Wi-Fi technology uses radio for communication, typically operating at a frequency of 2.4GHz. Electronics that are "Wi-Fi Certified" are guaranteed to interoperate with each other regardless of brand. Wi-Fi is a technology designed to provide to the lightweight computing systems of the future, which are mobile and designed to consume minimal power. PDAs, laptops, and various accessories are designed to be Wi-Fi compatible. There are also phones under development which would switch seamlessly from cellular networks to Wi-Fi networks without dropping a call that is also the aim of this thesis.

Most recently, merging Wi-Fi with VoIP is a hot topic in the wireless communication. It is called VoWiFi (**V**oice **o**ver **W**ireless **F**idelity) simply means a Wi-Fi based VoIP service or in more general words, a wireless based VoIP system which can work on PDA or multimedia handsets. VoWiFi decreases the communications cost while having a mobile system that offers more reliable coverage indoors and higher voice quality than traditional cellular service.

2.4.2. GSM, 3G

3G or the third-generation wireless technology represents the convergence of various 2G wireless telecommunications systems into a single uniform global system which includes terrestrial and satellite components in its functioning. 3G cellular network refers to near future developments in personal and business wireless technology, especially relating to mobile communications. 3G introduces many benefits as roaming capability, broad bandwidth and high speed communication (upwards of 2Mbps).

The 3GPP (Third Generation Partnership Project) is producing globally applicable Technical Specifications and Technical Reports for third generation mobile systems. The group is using IP technology end-to-end to deliver multimedia content to mobile handsets; therefore IETF protocols are needed. In this project, SIP has been selected as the signalling protocol for 3G networks. Every call made in a 3G network is established using SIP.

Users are identified by SIP URLs and/or E.164 numbers which is the numbering system of the telephone system. The carrier system (GPRS or mobile IP) manages *micro-mobility* which is the movement of the mobile user from one base station to another. *Macro-mobility*, which is the movement of the mobile user from one domain to another, is handled by SIP. SIP routes signalling so that services are available from the originating or terminating network.

3GPP has identified the Call State Control Function (CSCF) in the network. This is the equivalent of a SIP server. There will be three different kinds of CSCF:

- Proxy CSCF - this is the first point of contact in a visited network and finds the user's home network and provides some translation, security and authorization functions.
- Serving CSCF - controls sessions, acts as registrar and triggers and executes services. The serving CSCF accesses the user's profile. It can be located in the home network.

Interrogating CSCF is the first point of contact in the home network. It assigns the serving CSCF, contacts the Home Subscriber Server (HSS) and forwards SIP request.

3G networks are designed to enable rich voice and data applications – such as multimedia messaging service, gaming and mobile Internet – that are integrated, interactive and presence-aware. This means that 3G is tailor made for SIP. According to the ITU specification for 3G, it should support data rates of 144 kbps for mobile users, 384 kbps for pedestrians, and 2 Mbps for fixed applications.

2.5. Wireless Devices Technologies

A number of new communications devices have appeared over the last decade, including the phone and the PDA. These devices make people increasingly independent within their work and leisure environments. Because networks are now converging on an IP infrastructure, it is becoming possible to make sense of these different forms of communication. Also, since SIP client software is lightweight, it can be embedded in all these different devices bringing them under a single umbrella so that services can cross all platforms [24]. Using SIP as the signalling protocol means that sessions can be established between different devices which negotiate the appropriate media capability. The devices become means of accessing those services associated with a user.

A simple example of this is single number reachability. Due to presence capability of SIP, at any time the network knows where a user is. This means that he/she is accessible and available to take a call. The network will divert any incoming calls to the correct device automatically.

On the other hand, there are some limitations to services that can be related to bandwidth and screen size of the devices. Bandwidth problems can be eliminated with the approach of 3G networks. Also, handsets can display information in an appropriate format.

2.5.1. Java and J2ME

Java is a recently developed concurrent, class-based, and object oriented programming language which provides the basic object technology of C++ with some enhancements and some deletions. Java source code is compiled into an architecture independent object code. The object code is interpreted by a Java runtime system. With Java technology, you can use the same application on any kind of machine such as a PC, a Macintosh computer, a network computer, or even new technologies like Internet screen phones.

The Java 2 Platform comprises three elements as explained in [27]:

- The *Java programming language* is syntactically similar to C++ but differs fundamentally. While C++ uses unsafe pointers and programmers are responsible for allocating and freeing memory, the Java programming language uses type safe object references, and unused memory is reclaimed automatically. Furthermore, the Java programming language avoids multiple inheritances (a likely source of confusion and ambiguity in C++) in favour of a cleaner construct, *interfaces*.

- A *virtual machine* forms the foundation of the Java platform. This architecture offers several attractive features: The virtual machine can be implemented to run a top a variety of operating systems and hardware, with binary-compatible Java applications operating consistently across many implementations. In addition, the virtual machine provides tight control of executed binaries, enabling safe execution of un-trusted code.
- Finally, *an extensive set of standard application programming interfaces (APIs)* rounds out the Java platform. These support almost everything you might want your applications to do, from user interface through cryptography, from CORBA connectivity through internationalization.

The Java language, Java virtual machine, and Java APIs together compose the Java platform. Moreover, the Java platform is designed to encompass a wide range of computer hardware, everything from smart cards through enterprise servers.

The Java 2 Platform, Micro Edition (J2ME) has been developed primarily as a technology for the execution of applications on small, limited memory devices [28]. In this case, these small, constrained devices are mobile phones, PDAs, TV set-top boxes, in-vehicle telemetry, residential gateways and other embedded devices. Even though J2ME is targeted at devices with limited capabilities, it has been derived from J2SE and shows all the characteristics of the Java language.

Portability of an application and service code across various J2ME-based devices is insured through the use of J2ME *Configurations* and *Profiles* [29]. Basically, a *configuration* is a specification. A J2ME configuration specification is aimed at devices with similar requirements of memory size and processing power. To ensure portability across the device range, configurations cannot contain any optional features. A configuration defines the minimum Java libraries and VM capabilities that a developer can expect to find on all mobile devices implementing the configuration specification. A *profile* is a collection of Java technology APIs that supplement a configuration to provide capabilities for a specific device or market type.

2.5.2. Symbian OS and S60 Platform

Symbian OS is a *proprietary* operating system, designed for mobile devices, with associated libraries, user interface frameworks and reference implementations of common tools, produced by Symbian Ltd. It is an open operating system that is specifically designed for data-enabled 2G, 2.5G and 3G mobile phones.

It includes multi-tasking kernel, communications protocols, integrated telephony support, advanced graphics support, low- level graphical user interface framework, etc [30]. However, as the current kernel of the Symbian OS is not real-time kernel, it needs some support from the underlying software to be able to support communication protocols like GSM and GPRS that have real-time properties. Normally real-time services are provided by a domestic operating system (DOS) that runs under Symbian OS.

There are multiple platforms, based upon Symbian OS that provide a software development kit for application developers working on a Symbian OS device. The main ones are UIQ and Series 60 (S60). UIQ and S60 are two user interfaces (UIs) available for Symbian OS and both provide a framework, built on top of Symbian OS, which can be reused by application writers, and a set of standard applications, for example, Personal Information Manager Applications like multimedia and email. In this thesis, S60 is used for the UI implementation of proposed VoIP client on mobile phones.

S60 is one of a range of developer platforms created by Nokia, and it has several versions. There have been three releases of S60: "Series 60" (2001), "Series 60 Second Edition" (2004) and "Series 60 3rd Edition" (2005). Versions 1.0, 1.1 and 1.2 (generally called as the Series 60 Platform 1.x) are based on Symbian OS v6.1. Series 60 Platform 2.0 is based on Symbian OS v7.0s. It is noteworthy that software written for S60 1st edition (S60v1) or 2nd edition (S60v2) is not binary compatible with S60 3rd edition (S60v3), because it uses a new, hardened version of the Symbian OS (v9.1).

The main features introduced in Platform 2.0 that affect application UIs are skins and bidirectional text support [31]. Skins allow users to customize the UI, by changing the background bitmap, icons and color scheme. Bidirectional text support allows languages that are written from right to left, for example, Hebrew and Arabic, to be edited and displayed. It also affects the ordering and alignment of controls throughout the UI.

All Series 60 phones use a navigation controller that allows navigation in four directions, a confirmation key, and two hardware buttons, called soft-keys, beneath the screen. These buttons make Series 60 phones easy to use with one hand. Users can input text using the phone's keypad and can optionally use a predictive text input system [31].

Java ME applications for Symbian OS are developed using standard techniques and tools such as the Sun Java Wireless Toolkit (formerly the J2ME Wireless Toolkit) which is explained in Section 2.5.1. They are packaged as JAR and possibly JAD files. There are some tools which can be used to build Symbian 7.0 and 7.0s programs using Java. S60 3rd Edition Feature Pack 1 phones have Symbian OS 9.2 that is the final release of Symbian.

3. SIP BASED MOBILE VOIP CLIENT SYSTEM ARCHITECTURE

3.1. Main Components

As we have mentioned early chapters, we have three general goals in the design of our VoIP client architecture. The first goal is to handle mobility of the client between different wireless access technologies and to hand over between different networks seamlessly. The second goal is to make the system adapt its communication type according to the characteristics of new network. The last purpose is to improve basic half-duplex communication (push-to-talk) model and design a new PTT communication flow. Therefore, the architecture of the proposed VoIP client system is basically divided into three main parts: (i) Call Setup Manager, (ii) SIP-based Seamless Handover Mechanism, and (iii) SIP-Specific Event-Based Push-to-Talk service.

In this chapter, we present the details of the components of our SIP-based Mobile VoIP client.

3.2. Call Setup Manager

The Call Setup Manager (CSM) provides a decision mechanism to set conversation model (Half-duplex/Full-Duplex). Decision mechanism considers the bandwidth of the currently connected network to decide the conversation model. The proposed client's decision mechanism has the knowledge of theoretical results of different networks' bandwidth. This knowledge base consisting of network bandwidth values is constant and not changed experimentally.

Table 3.1: Conversation Type Decision Matrix

Air Interface	GPRS/Edge	3G	Wi-Fi/WLAN
GPRS/Edge	Half-duplex	Half-duplex	Half-duplex
3G	Half-duplex	Full-duplex	Full-duplex
Wi-Fi/WLAN	Half-duplex	Full-duplex	Full-duplex

Table 3.1 shows decision mechanism of CSM according to bandwidth knowledge base. The first row and the first column are the air interfaces of the communication parties (the caller and the called clients). For instance, if the *caller* is connected via GPRS and the *called client* is connected via WLAN, the conversation model will be decided as half-duplex, since GPRS has less bandwidth than WLAN to support full-duplex communication.

If the *call imitator* has a capability of making a full-duplex conversation, it starts a normal SIP call procedure because at the beginning, it does not have knowledge about called client's network. If the *called client* does not have a full-duplex conversation capability, the *called client* sends "*302 Moved temporarily*" message and redirects the call to its own IP address again. In addition, it sets the "*redirection-reason*" field to PTT as shown in Figure 3.1, which means that the communication will be half-duplex.

SIP/2.0 **302 Moved temporarily**
From: "client1" <client1@itu.edu.tr>;tag=2
To: <sip:client2@itu.edu.tr;user=phone>;tag=3
Call-ID: 12345600@itu.edu.tr
Contact: sip:client2@itu.edu.tr;user=phone;**redirection-reason=PTT**
CSeq: 1 INVITE

Figure 3.1: Half-duplex conversation setup

If the *caller* has a capability of SIP-Specific Event-Based Push-to-Talk, which is explained in later of this chapter, it initiates a subscription process that is discussed in Section 3.4. If not, *caller client* sends a new "*INVITE*" message that has a "*redirection-reason*" field set to PTT as illustrated in the Figure 3.2:

INVITE sip:client2@itu.edu.tr;user=phone;**redirection-reason=PTT**
SIP/2.0
From: "client1" <client1@itu.edu.tr>;tag=2
To: <sip:client2@itu.edu.tr;user=phone>
Call-ID: 12345600@itu.edu.tr
CSeq: 1 INVITE

Figure 3.2: Basic PTT communication setup

In this case, the *called client* looks a subscription list. If the *caller* is not found in the list, the call will be rejected. If it is found in the list, it continues with the PTT initialization procedure that is discussed in Section 3.4.

3.3. Seamless Handover Management Model

The SIP-based Seamless Handover Mechanism (SHM) is responsible for the registration procedure of the new network and to change existing conversion session into a new conservation session. SHM use the same knowledge base of bandwidths discussed in CSM in order to decide handover. SHM knows all network interfaces on the device and has a capability of distinguishing messages from different interfaces and distributing messages to the correct interfaces. Indeed, whole procedure of the detection mechanism is dependent on operating system (OS) capabilities. OS of the client must support to manage multiple network interfaces. Basically, SHM stores all active network interfaces in a *listening point list*. However, in our VoIP client, the user can define his preferences. It means that the user can specify the network interfaces that he wants to be connected while he is talking. If a device has a network interface that is not activated yet, it will not be stored in the *listening point list*. If it is activated and is in the user defined networks list, it will be added to the *listening point list*.

SHM is a periodic sleep/listen process. It is started when the client application is initialized, and it is re-invoked every one minute if there is a no active call on the client. If SHM finds better network conditions during no active call states, it simply changes the network as shown in Figure 3.3. If the client does not connect to any SIP server, registration procedure in the figure is omitted.

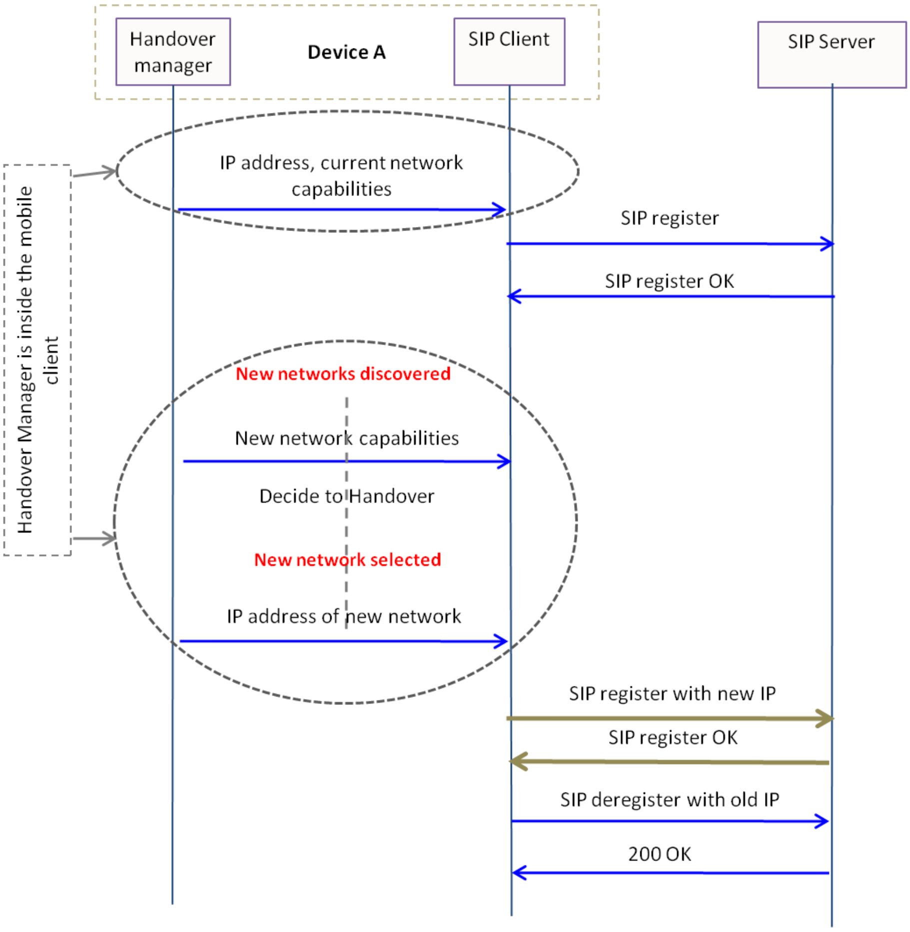

Figure 3.3: Handover procedure for no active call state

For active call states, SHM's sleep/listen period changes according to the conversation type of the call. These conversation types are half-duplex and full-duplex. If the active call is half-duplex, SHM is re-invoked every 30 seconds. When SHM finds better network conditions, it starts the seamless handover procedure as illustrated in Figure 3.4. If the call is full-duplex, but network capabilities are not highest, SHM is invoked every 20 seconds. For instance, WLAN is generally the cheapest and has more bandwidth than 3G cellular network; in this case, SHM tries to increase quality of service.

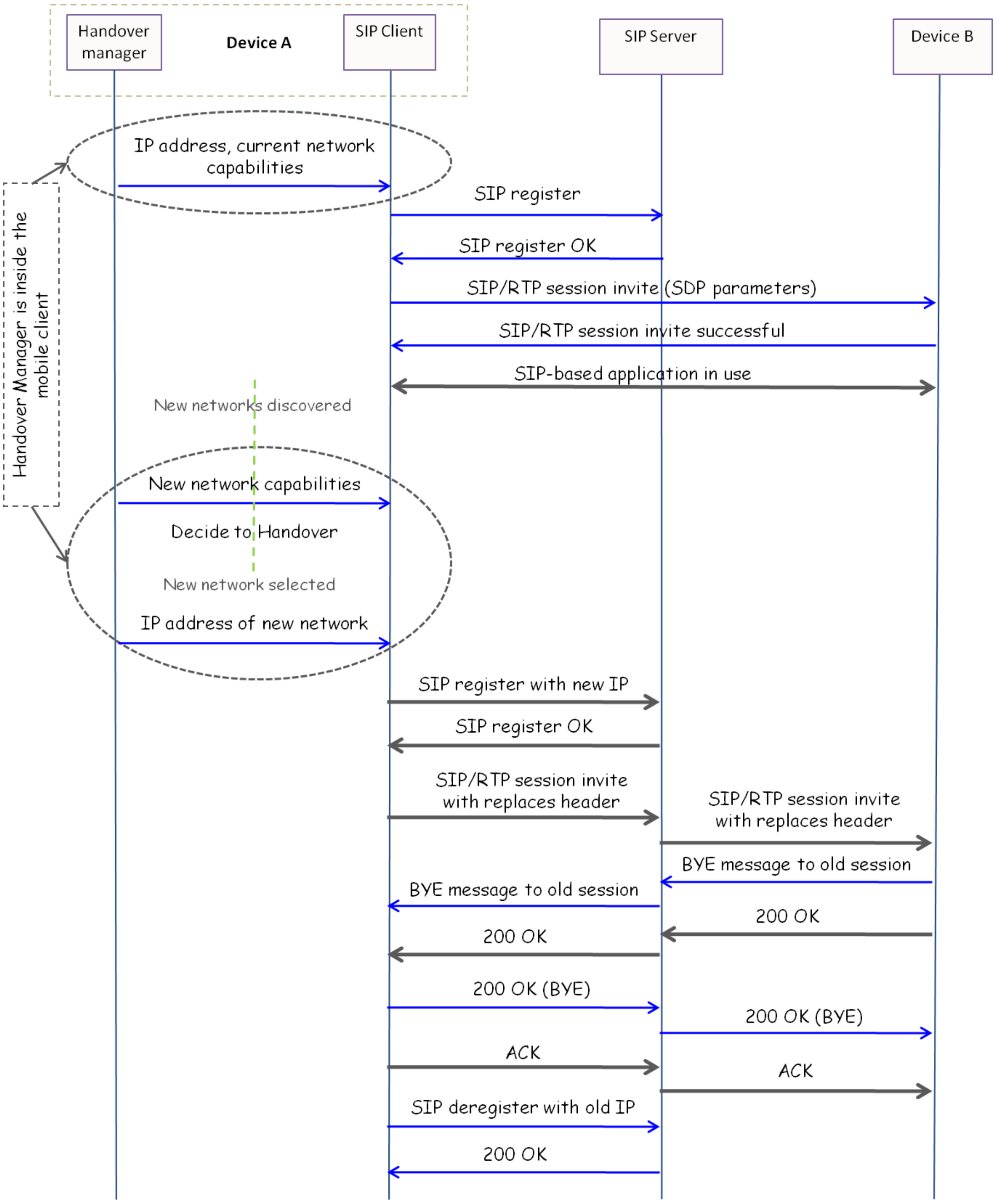

Figure 3.4: Handover procedure during an active call

When the client is connected to the SIP server, SHM adds a network connection interface into the *listening point list*. Then, SIP client sends a SIP "*REGISTER*" message to the SIP server and waits for the response. If the server returns "*200 OK*" message, it means that registration is completed successfully.

When a new network is detected during the active call, operating system is triggered to connect to this new network. The new IP is passed to the SIP client's registration service and SIP client registration service sends new “*REGISTER*” message to the server and waits for the “*200 OK*” message. While this operation is in progress, client side has two valid open registrations. This means that both the old and the new IP addresses are valid for a while after registration of new IP. However, only return responses use the new registration.

The caller SIP client sends a new “*INVITE*” message with the “*replaces header*”. Also, the caller SIP client knows the previous session SDP information, so it creates a new SDP using the session information used before. Therefore, caller client adds a smaller SDP data into this “*INVITE*” message in order to use bandwidth efficiently. Furthermore, “*replaces header*” information is taken from the SHM. The “*replaces header*” contains *call-ID*, *to tag* and *from tag* information. The “*replaces header*” provides to change the old session with the new session.

When destination client receives a new “*INVITE*” message with a “*replaces header*”, it checks its supported headers. If it does not support “*replaces header*”, it returns a “*420 Bad extension*” message. If the destination client supports “*replaces header*”, it will start to search all the active sessions. If it does not find the session, it returns a “*481 Dialog Does Not Exist*” message. If the session exists, the destination client sends a “*BYE*” message to the old session and a “*200 OK*” message to the new session. In this period, destination client creates RTP session and waits for the first RTP message. When the SIP client, which initiates the handover, receives the “*200 OK*” message, it creates a new RTP session and sends the data packets over this RTP session, then stops the old session. Finally, it sends an “*ACK*” message. If the destination client does not receive any “*ACK*” message in 60 seconds, even if it receives the RTP packet, the destination client sends a “*BYE*” message and closes the session. After the “ACK” message is received, destination RTP stream sends and receives voice packets. Finally, the SIP client, who performs the handover, sends a new “*REGISTER*” message with the “*expires header*” equal to zero (un-registration) by using the old IP address.

The main difference between the IETF proposed SIP mobility model and our SIP-based handover model is that “*RE-INVITE*” message is not used to change the session in our model. Instead, a new “*INVITE*” message with “replaces header”, which identifies the existing session, is used. Therefore, the new “*INVITE*” message does not have to follow the previous session’s “*INVITE*” message routing path. This means that the new session routing path only uses the new wireless network routing path, and does not require visiting old wireless network routing path anymore.

In addition, our SIP-based handover model starts RTP session initiation without accepted SIP established session. Thus, RTP session is changed using "*INVITE*" and "*200 OK*" messages, but "*ACK*" message still require completion of SIP call setup.

3.4. SIP-Specific Event-Based Push-To-Talk

PTT service is responsible to manage half-duplex conversion setup. This service can be provided by a client or a server application. In our proposed model, media channel is controlled using the SIP event notifications mechanism which is discussed in Section 2.1.4. Therefore, RTP session is independent from any intercommunication server, so that connection can be established directly, and intercommunication server delays are eliminated.

3.4.1. Client to Client Model

In this model, SIP event notifications (see Section 2.1.4) are managed by the called SIP client. When the caller wants to talk over PTT, the caller has to subscribe to the called client's PTT service. The whole subscription process for the Client-to-Client

PTT is depicted in Figure 3.6. Firstly, the caller client (Client A) sends subscribe message to the called client (Client B) like in Figure 3.5:

SUBSCRIBE sip:clientB@itu.edu.tr SIP/2.0
Via: SIP/2.0/UDP sis.itu.edu.tr;branch=z9hG4bKnashds7
From: clientA @ itu.edu.tr;tag=123s8a
To: clientB@ itu.edu.tr
Call-ID: 9987@ itu.edu.tr.com
Max-Forwards: 70
CSeq: 9887 SUBSCRIBE
Contact: clientA @ itu.edu.tr
Event: PTT

Figure 3.5: PTT Subscription message from Client A to Client B

Destination client looks the "*Event header*" to distinguish the subscription. In this case, the PTT service subscription's "*Event header*" is PTT; therefore, the destination client understands that Client A wants to subscribe its own PTT service. If the Client B provides this service, it returns the "*200 OK*" message. After that, Client A becomes the chairman of this conversation. Furthermore, Client B adds Client A into its subscription list and Client B does not accept any PTT request outside of this list's members. When Client A sends the INVITE message, it sets the "*reason header*" to PTT, so that Client A indicates the conversation type as a half-duplex. In addition, Client A puts "*send-only*" attribute into SDP part. Thus, Client A gets the first talk right. This message optimization reduces the call setup duration.

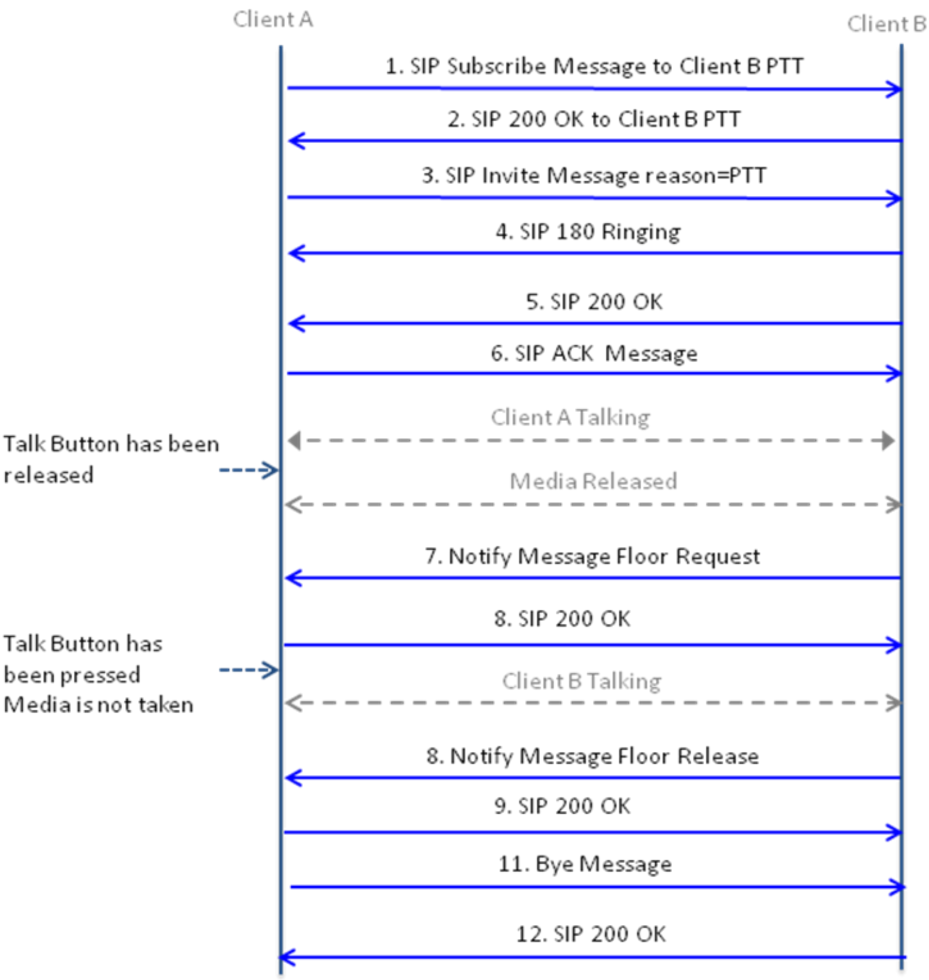

Figure 3.6: SIP-Specific Event-based Client-to-Client PTT procedure

When Client A releases the button, its application does not send a "*NOTIFY*" message because it is the chairman. It only stops the own RTP stream and sets the channel status as idle.

NOTIFY sip:clientA@itu.edu.tr SIP/2.0
Via SIP/2.0/UDP sis.itu.edu.tr:5060;branch=z9hG4bK3841323
Max-Forwards: 70
To: ClientA <sip:clientA@ itu.edu.tr >;tag=1814
From: <sip:clientB@itu.edu.tr >;tag=5363956k
Call-ID: 452k59252058dkfj34924lk34
CSeq: 1 NOTIFY
Contact: <sip:clientB@ itu.edu.tr >
Event: PTT
Subscription-State: active
Content-Type: application/xml+PTT
Content-Length: 49

<PttRequest>
<Ptt-action>request</Ptt-action>
</PttRequest>

Figure 3.7: Notify message from Client A to release media channel

If Client B presses the talk button, it sends a "*NOTIFY*" message with PTT *floor take* content as shown in Figure 3.7. When, Client A receives the "*NOTIFY*" message, it checks the channel status. If the channel status is idle, it returns "*200 OK*" message with "*200 OK*" content. If Client A presses the talk button, client application will not send any message, because chairman of this conversation is itself. Client application looks the media channel status. If the channel status is idle, it will give the right of sending media packets to the own service and change media channel status as a floor taken. If media channel status is not idle, Client A does not take the right of sending media packets to the network. If Client B sends a "NOTIFY" message, after Client A takes the media channel, Client A application returns "*200 OK*" message with "*486 Busy Here*" message body. Therefore, Client B is not permitted to take media channel to send data packets. If Client B releases the button, Client B application sends a "NOTIFY" message with PTT floor release content. When Client A receives the "NOTIFY" message, it sets the channel status as idle and returns the "*200 OK*" message with "*200 OK*" content. To close the conversion, "*BYE*" message is sent to other client. If the caller sends "*BYE*" message, it sends "*SUBSCRIBE*" message with expires value zero in order to unsubscribe from the called client's PTT service. If the caller client receives "*BYE*" message, it will send "*SUBSCRIBE*" message with expires value zero again.

3.4.2. Server-Based Model

Generally, SIP clients are connected to a SIP server to increase the service quality, and to decrease the client side load while providing privacy for the client. Therefore, it is important to provide same service from the server side.

Server side solution which is illustrated in Figure 3.8 does not change any logic of the client-to-client solution. Thus, the client, not connected to the any SIP server, do not realize a server as a service provider. Server side takes the responsibilities of subscription process, but media channel control is still controlled by the caller client's PTT service and also the RTP session is still end-to-end. When the Client A (caller) wants to talk with Client B (called) via PTT service, it will subscribe to Client B's PTT service as a client solution. When server receives the "*SUBSCRIBE*" message, it will check the destination client registration. If the called client registers to its server, its server will return "*200 OK*" message with "*200 OK*" body. After then, Client A sends "*INVITE*" message to start PTT. When server takes the "*INVITE*" message with "*reason header*" as PTT, it will search for the related user list. If the caller client is found in the list, it passes the "*INVITE*" message to the client. If caller is not found in the list, "*INVITE*" message sent by caller will be rejected by the server application and will not be sent to the client application. Moreover, servers have to pass "*reason header*" without changing its value, because the client learns the conversation type by looking at the value of "*reason header*".

If the caller client adds "*send-only*" attribute into SDP data of the "*INVITE*" message, it automatically takes the media channel to send data packet. When Client A releases the media channel, it does not send "*NOTIFY*" message with "*floor release*" content. It is locally handled by the application side request. If Client B releases the media stream, it sends "*NOTIFY*" message and this message will be delivered to the Client A because client side is responsible for the management of media channel. Client A handles this message and returns the appropriate response.

The main difference between server-based and client-based solution is that, SIP clients PTT service subscription is managed by their own servers. When a SIP client is registered to its SIP server, SIP server sends "*SUBSCRIBE*" message to its own registered client in order to send "*NOTIFY*" messages which will be received from caller client. In addition, SIP server adds the registered client to its own PTT service list.

When the client wants to stop PTT conversation, it has to send "*BYE*" message. If the using PTT service does not belong to the client, client has to send "*SUBSCRIBE*" message with "*expires header*" set to zero. The other client will not be unregistered from PTT service until it logs out. Therefore, SIP server deletes the records coming from the related user's PTT service list.

SIP Event Base Push to Talk (Server Solution)

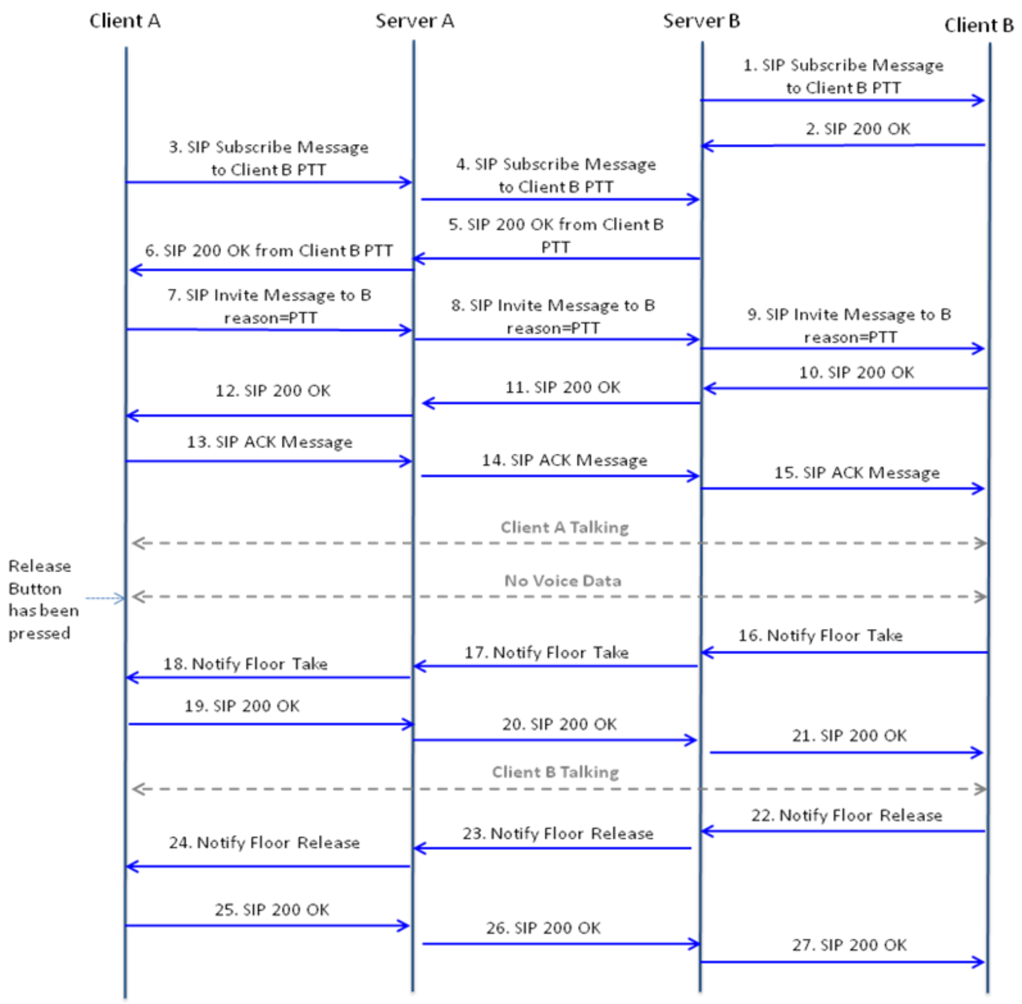

Figure 3.8: SIP-Specific Event-based PTT procedure for Server-based solution

4. IMPLEMENTATION OF SIP BASED MOBILE VOIP CLIENT

The implementation of SIP based Mobile VoIP client is divided into four main modules. These are (i) Call Setup Manager, (ii) Seamless Handover Management Model, (iii) SIP-Specific Event-Based Push-to-Talk, and (iv) User Interface and Database Management.

4.1. Call Setup Manager

Call Setup Manager is directly related to the OS system network capabilities. OS has to support management of multiple network interfaces. For example, JAVA based applications do not know which network interface is used, because JAVA does not give permission to manage network interfaces. Therefore, Symbian OS is preferred for implementation of the Call Setup Manager.

Call Setup Manager knows the currently used network interface properties using Symbian *RConnectionMonitor* API. In addition, Symbian OS defines *EBearerEdgeGPRS*, *EBearerWLAN*, *EBearerWCDMA*, *EBearerCDMA2000* enumerations for defining network types. *EBearerEdgeGPRS* shows 2.5G, *EBearerWLAN* shows WLAN and *EBearerWCDMA*, *EBearerCDMA2000* shows 3G networks. Thus, CSM uses the knowledge base about networks types and decides the conversation as a half-duplex or full-duplex. Then, this information is passed to the SIP call control service.

4.2. Seamless Handover Management Model

Another important property of Symbian OS is multi-task support, so client application and network searching mechanism can work at a same time. When the SIP client is initiated, Seamless Handover Management is started as a new thread, and then this thread is invoked every minute to search for available WLAN and Wi-Fi networks. If the client has an active call, SIP call control service changes the invoke period according to conversation type. If call is half-duplex, SIP call control service changes invoke period to 30 seconds and if call is full-duplex, SIP call control service changes invoke time to 20 seconds. Moreover, If the client is connected to highest network capabilities (ex: WLAN), SHM searching mechanism is stopped until the current network is disconnected.

When the new network is found, SHM gathers the network information and looks the knowledge base about network interfaces. If the new network capabilities are better than currently used network capabilities, it invokes the SIP application register service to start new registration procedure. Furthermore, SHM commits the message distribution during handover. SIP register service gets the information about which interface should be used to send new SIP messages. Therefore, registration messages for new network are sent over the new network connection interface and registration messages for old network are sent over the old network connection interface. It means that the old session does not change any routing route and the new session is independent from the old session routing path.

For socket connection, *RConnection* API is used. *RConnection* objects are implemented as sub-sessions to ESOCK (Socket Server for Symbian). ESOCK provides a generic interface to communications protocols through communication end points known as sockets. Therefore, SIP application does not know anything about network type in order to sent message. Moreover, *RConectionMonitor* API is used for finding the WLAN and Wi-Fi networks. Service Set Identifier (SSID) for WLAN and WIFI is found using this API.

4.3. SIP-Specific Event-Based Push-To-Talk

Firstly, SIP stack is initialized and basic services are registered to the SIP stack, so when the message is received, SIP stack knows which service will handle received message. Basic services are *register service*, *call control service*, and *subscribe service*. *Register service* is responsible for SIP server registration; *call control service* is responsible for call setup and hang-up operations while *subscribe service* is responsible for subscription to define events and to listen for event changes. SIP specific Event-base Push-to-Talk is not a basic SIP service. It uses *call control* and *subscribe services*.

Subscribe service manages subscription and notification procedures. If it receives the "*SUBSCRIBE*" message from the other client, it adds the client to related event's watcher list. If any changes are occurred about subscribed event, subscription service sends "*NOTIFY*" message to the clients in order to notify the change. PTT service does not keep the clients (responsibility of subscribe service), it only knows the media channel status and media channel chairman.

For PTT service, "*NOTIFY*" messages are used for floor controlling, so that it is possible that watcher client sends "*NOTIFY*" message about floor control request. When subscribe service receives the "*NOTIFY*" message, it searches the watcher list. If the client is not in the list, "*NOTIFY*" message is discarded. If the client is in the list, request is sent to PTT service. PTT service decides to response and returns the response message to the subscribe service. Subscribe service returns the response to the client.

When "*INVITE*" message is received with a "*reason header*" which equals PTT, call setup service triggers the PTT service. PTT service queries the subscribe service's PTT event list. If the client is not found, call is rejected. Otherwise, call setup service takes the responsibility of call initiation, but SDP data information is still supplied by a PTT service.

PTT service uses FIFO (First In First Out) algorithm to decide the media channel control mechanism. When the "*NOTIFY*" request is sent to PTT, PTT service locks the status object and control the status. If status is idle and request is floor request, status is changed to floor taken and status object lock is removed. If the status is taken and request is floor release, status is changed to idle and status object lock is removed. For both condition, PTT service returns "*200 OK*" response. If the status is taken and request is floor request, status is not changed and status object lock is removed. PTT service returns error code.

4.4. User Interface and Database Management

SIP services, SIP stack and database persistence objects are independent from the SIP client user interface. User interface classes know the SIP services, database persistence objects, and data access objects.

Login UI, a*pplication UI*, *call UI* and *contact UI* are the main components of the SIP client. *Login UI* gets the login information from the user. This information contains username, password, and server IP and server port. If the client does not connect any SIP server, only username and password are enough. *Application UI* contains the application menu and contact list. The called user can be chosen from contact list. Contact list is created using the *contact UI*. *Contact UI* stores the address and nickname information. User can use the menu and press the call item in order to call undefined client. *Call UI* uses only called user URI to start call set-up. *Call UI* supports call operations, hold and hang-up; and push-to-talk operations like talk, release and hang-up operations.

UI classes use Database Access Objects (DAO) to access database. These objects use the persistence objects to manipulate data. Our client only needs three DAO object and three persistence object. These DAO objects are *SIPServersDao*, *LoginDao* and *ContactUserDao*. Each DAO object supports save, delete, update and query operations. Results sets are converted to the *SIPServer*, *ContactUser* and *Login* persistence objects. *SIPServer* persistence object keeps server IP and server port information. Contact address, port, nickname, and order information are kept in *ContactUser* persistence object. *Login* persistence object keeps login name, password and display name information.

5. CONCLUSIONS AND FUTURE WORK

Over the years, several schemes have been proposed for mobility management in IP networks. Mobile IP is as a network layer protocol for seamless mobility across heterogeneous networks has gained momentum in the last years. However, for real-time applications, which are generally delay-intolerant applications, such as VoIP, mobile IP still has shortcomings in terms of providing a means of fast handover with minimal or no packet loss. In this thesis, we have proposed an application layer scheme at the client side that works under heterogeneous air interfaces and their corresponding networks. It basically enables "*make-before-break*" handover between heterogeneous networks by employing both old and new IP addresses during the handover process.

Therefore, the aim of this thesis is to design a VoIP application for mobile devices that is capable of talking anywhere and anytime. The proposed SIP-based Mobile VoIP client for wireless networks has been achieved this goal because it provides seamless handover among different access networks. Moreover, it gives the users the opportunity of changing conversation type from full-duplex to half-duplex or vice versa according to bandwidth of communication networks to achieve best effort call quality during a call. The VoIP application presented in this thesis, also, provides a better PTT communication flow which decreases the communication delays created by internal servers.

As future works, we can basically mention about two issues, security and QoS of the communication, which are also needed for a VoIP application.

SIP provides a suite of security services, which include denial-of-service prevention, authentication (both user to user and proxy to user), integrity protection, and encryption and privacy services. The security related issues and their implementations are not included in this thesis; however, enabling these SIP capabilities like encryption and security is left as future work.

Secondly, SIP needs some enhancements for the interworking with a QoS enabled network. Therefore, the interaction between SIP and resource management for QoS should be studied as well. This open issue covers the problem of how to setup QoS before alerting the user, in order to avoid that the user answer the call but resources are not available. To develop a QoS aware SIP application, RTP QoS feedbacks can be used. Moreover, the end-to-end communication delay should be evaluated and the application should be tested under different communication networks to see the performance in different conditions.

REFERENCES

[1] **Perkins, C.**, 1997. Mobile IP, *IEEE Communications Magazine*, May 1997, Vol. **35**, pp. 84–99.

[2] **Wedlund, E. and Schulzrinne, H.**, 1999. Mobility Support using SIP, *Proceedings of the 2nd ACM International Workshop on Wireless Mobile Multimedia*, 1999, pp. 76 - 82.

[3] **Johnston, A. B.**, 2004. Chapter 9: Wireless and 3GPP, in *SIP - Understanding the Session Initiation Protocol*, London, Artech House, pp. 194 - 201.

[4] **Pangalos, P.A., P., Boukis, K., Burness, L., Brookland, A., Beauchamps, C., & Aghvami, A.**, 2001. End-to-End SIP Based Real Time Application Adaptation During Unplanned Vertical Handovers, *Proceedings of IEEE Global Telecommunications Conference*, Vol. **6**, pp. 3488-3493.

[5] **Jung, J.W., Mudumbai, R., Montgomery, D., & Kahng, H.,** 2003. Performance Evaluation of Two Layered Mobility Management using Mobile IP and Session Initiation Protocol, *Proceedings of IEEE Global Telecommunications Conference*, Vol. **3**, pp. 1190- 1194.

[6] **Dutta, A., Madhani, S. and Chen, W.**, 2004. Fast-Handoff Schemes for Application Layer Mobility Management, *Symposium of IEEE Personal, Indoor and Mobile Radio Communications, Spain*.

[7] **Sattari, N., Pangalos, P. and Aghvami, H.**, 2004. Seamless Handover between WLAN and UMTS, *Proceedings of IEEE Vehicular Technology Conference*, Vol. **5**, pp. 3035- 3038.

[8] **Rajkumar, A., Feder, P., Benno, S., & Janiszewski, T.**, 2004**.** Seamless SIP-Based VoIP in Disparate Wireless Systems and Networks, *Bell Labs Technical Journal*, pp. 65–82.

[9] **Bellavista, P., Corradi, A. and Foschini, L.**, 2006. SIP-Based Proactive Handoff Management for Session Continuity in the Wireless Internet, *Proceedings of*

the 26th IEEE International Conference on Distributed Computing Systems Workshops, Portugal, p. 69.

[10] **Cardenete-Suriol, M., Mangues-Bafalluy, J., Portoles-Comeras, M., Requena-Esteso, M., & Gorricho, M.**, 2007. VoIP performance in SIP-based vertical handovers between WLAN and GPRS/UMTS networks, *Proceedings of IEEE International Conference on Communications (ICC '07)*, pp. 1973-1978.

[11] **Choong, K., Kesavan, V., Ng, S., Carvalho, F., Low, A., & Maciocco, C.**, 2007. SIP-based IEEE802.21 media independent handover — a BT Intel collaboration, *BT Technology Journal*, Vol. **25**, No. 2, pp. 219 - 230.

[12] **Parthasarathy, A.**, 2005. Push to Talk over Cellular (PoC) Server, *Proceedings of IEEE Networking, Sensing and Control*, March 2005, pp. 772 - 776.

[13] **Raktale, S.K.**, 2005. 3Poc : An Architecture for Enabling Push To Talk Services in 3GPP Networks, *Proceedings of IEEE International Conference of Personal Wireless Communications*, pp. 202 - 206.

[14] **Kim, P., Balazs, A., Broek, E., Kieselmann, G., & Böhm, W.**, 2005. IMS-based Push-to-Talk over GPRS/UMTS, *Proceedings of IEEE Wireless Communications and Networking Conference*, Vol. **4**, pp. 2472- 2477.

[15] **Peng, C. and Xuejun, Y.**, 2005. Performance Analysis of SIP-based Push-to-Talk Service for GPRS/cdma2000 Network, *Proceedings of IEEE Mobile Technology, Applications and Systems*, November 2005, p. 4.

[16] **Blum, N. and Magedanz, T.**, 2005. PTT + IMS = PTM - Towards Community/Presence-based IMS Multimedia Services, *Proceeding of IEEE International Symposium on Multimedia*, pp. 337-344.

[17] **Menkens, C., Kjellin, N. and Davoust, A.**, 2007. IMS Social Network Application with J2ME compatible Push-To-Talk Service, *Proceedings of IEEE Next Generation Mobile Applications, Services and Technologies*, pp. 70-75.

[18] **Rosenberg, J., Schuluinne, H., Camarilla, G., Johnston, A., Sparks, R., Handley, M., et al.**, 2002. SIP: Session Initiation Protocol, [RFC 3261, Updated by RFCs 3265, 3853.], The Internet Engineering Task Force, June 2002.

[19] **Glasmann, J., Kellerer, W. and Müller, H.**, 2001. Service Development and Deployment in H.323 and SIP, *Sixth IEEE Symposium on Computers and Communications*, Tunisia, pp. 378-385.

[20] **Andreasen, E. and Foste, B.**, 2003. Media Gateway Control Protocol (MGCP) Version 1.0., *IETF Network Working Group,* USA, January 2003, pp. 5-33.

[21] **Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V.**, 2003. RTP: A Transport Protocol for Real-Time Application, in *RFC 3550*, *IETF Network Working Group*, USA, July 2003.

[22] Free Skype VoIP Systems & VoIP Service. [Online] [as of 11 13, 2007.] http://skype.free.net.pl.

[23] **Roach, A. B.,** 2002. Session initiation protocol (SIP)-specific event notification, in *RFC 3265, Internet Engineering Task Force*.

[24] What is SIP Introduction / About SIP. [Online] [as of 11 13, 2007.] http://www.sipcenter.com.

[25] **Handley, M., Jacobson, V. and Perkins, C.**, 2006. SDP: Session Description Protocol, in *RFC 4566, Internet Engineering Task Force,* July 2006.

[26] **Audio-Video Transport Working Group, Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V.**, 1996. RTP: A Transport Protocol for Real-Time Applications, in *RFC 1889*, *Internet Engineering Task Force*.

[27] Introduction to Mobility Java Technology. *Sun Mobile Device Technology*. [Online] [as of 12 18, 2007.] http://developers.sun.com/mobility/getstart/.

[28] **Jode, Martin de.**, 2004. Chapter 1: Introduction to J2ME, in *Programming Java 2 Micro Edition on Symbian OS: A developer's guide to MIDP 2.0.*, Symbian Press.

[29] **Day, Bill.**, 2001. Developing Wireless Applications using the Java 2 Platform, Micro Edition, *Sun Microsystems, Inc.*, August 2001.

[30] **Sukanen, Jari.**, 2004. Extension framework for Symbian OS applications. Helsinki, *Master Thesis*, Helsinki University of Technology, Department of Computer Science and Engineering.

[31] **Harrison, Richard.**, 2004. Chapter 2: Symbian OS User Interfaces, in *Symbian OS C++ for Mobile Phones: Programming with Extended Functionality and Advanced Features,* Symbian Press.

RESUME

Kamil ERMAN was born in Burdur, in 1981. He completed his higher education in Burdur Cumhuriyet Super Lycee, in 1999. He received his Bachelor of Science degree from Yeditepe University Computer Engineering in 2004. Same year, he is accepted to Istanbul Technical University Computer Engineering for Master programme. He worked in Telenity Signalling and Media Department as a software engineer, between 2004 and 2006. Then, he worked in Nortel Netas MultiMedia Communication department as a software design engineer between 2006 and 2007. Currently, he works in Sqills B.V. in Netherlands as a Senior Application Developer.