CERTIFICATE

I, MEENAKSHI hereby certify that the work which is being presented in this thesis entitled **"IMPLEMENTATION OF INSTANT MESSAGING & PRESENCE SERVICES IN IP MULTIMEDIA SUBSYSTEMS**" by me in partial fulfillment of requirements for the award of degree of Master of Engineering in Electronics and Communication from THAPAR INSTITUTE OF ENGG & TECH (Deemed University), PATIALA, is an authentic record of my own work carried under the supervision of Mr. RAJESH KHANNA and Mr. BALWANT SINGH at TIET, PATIALA.

The matter presented in this thesis has not been submitted in any other University or Institute for the award of any degree.

(**MEENAKSHI**)
Signature of the student

This is certified that the above statement made by the candidate is correct to the best of my knowledge.

(Mr. RAJESH KHANNA)
SUPERVISOR &
ASSISTANT PROFESSOR, TIET PATIALA

(Mr. BALWANT SINGH)
SUPERVISOR &
SR LECTURER, TIET PATIALA

(Dr. R.S. KALER)
Head of Department, ECED
T.I.E.T, PATIALA.

(Dr. T.P. SINGH)
Dean of Academic Affairs,
T.I.E.T, PATIALA.

ACKNOWLEGEMENT

Words are often too less to reveal one's deep regards. An understanding of the work like this is never the outcome of the efforts of a single person. I take this opportunity to express my profound sense of gratitude and respect to all those who helped me through the duration of this thesis.

First, I would like to thank the Supreme Power, the GOD, who guided me to work on the right path of life. Without his grace, this would not have been possible. This work would not have been possible without the encouragement and able guidance of **'Mr. RAJESH KHANNA'**, Assistant Professor and **'Mr. BALWANT SINGH',** Sr Lecturer, TIET, PATIALA. Their enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. His feedback and editorial comments were also invaluable for the writing of this thesis. I am grateful to **Head of the Department Dr. R. S. KALER** and **Dr. A.K. CHATTERJEE** for providing the facilities for the completion of thesis.

I am also thankful to **Mr. SAMEER BHATIA**, I.T.A, Tata Consultancy Services Ltd., Gurgaon and **Ms. IRA ACHARYA**, Group Leader, Tata Consultancy Services Ltd., Gurgaon for their guidance and support throughout the thesis work.

I take pride of my self being daughter of ideal great **PARENTS** whose everlasting desire, sacrifice, affectionate blessing and help without which it would have not been possible for me to complete my studies.

I am highly indebted to my husband **Mr. SANJEEV SINGLA** without whose inspiration and constant help, it would not have been possible to complete this work.

At last, I would like to thank all the members and employees of Electronics and Communication Department, TIET Patiala whose love and affection made this possible.

(MEENAKSHI)

ABSTRACT

The IP Multi-Media Subsystem (IMS) is an IP multimedia and telephony core network. It is defined by 3GPP and 3GPP2 standards and organizations based on IETF Internet protocols. IMS is access independent. IMS permits and enhances real time, multimedia mobile services such as rich voice, video telephony, messaging, conferencing and push services by responding to the emerging trend to move toward a common, standardized subsystem. IMS represents a standardized, reusable platform providing a better way to experiment with, deploy, integrate, and expand consumer and enterprise voice and data services.

The next generation networks provided a new face to the telecommunication world. Earlier the main emphasis was on speech and speech related services, but nowadays the main aim is to enable faster data rates and various multimedia services. IP Multimedia Subsystems (IMS) provide a single platform for all the present as well as future technologies. IMS enables a whole new set of services such as instant messaging & presence services which is a novel concept of IMPS (Instant Messaging and Presence Service) brings instant messaging, one of the most rapidly growing Internet services, to mobile networks.

IMPS provide several advantages over conventional SMS messaging, including group messaging, presence information and conversations. IMPS is not limited to the mobile environment - it can provide a global service spanning both fixed and mobile subscribers. It includes client device availability (my phone is on/off, in a call), user status (available, unavailable, in a meeting), location, client device capabilities (voice, text, GPRS, multimedia) and searchable personal statuses such as mood (happy, angry) and hobbies (football, fishing, computing, dancing). Since presence information is personal, it is only made available according to the user's wishes - access control features put the control of the user presence information in the users' hands. Instant Messaging (IM) is a familiar concept in both the mobile and desktop worlds. Desktop IM clients, two-way SMS and two-way paging are all forms of Instant Messaging.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

3GPP	Third Generation Partnership Project
APEX	Application Exchange
BGCF	Breakout Gateway Control Function
CLP	Command Line Protocol
CSCF	Call Session Control Function
CPIM	Common Profile for Instant Messaging
CPP	Common Profile for Presence
CSP	Client-Server Protocol
DTD	Document Type Definition
GGSN	Gateway GPRS Support Node
GLMS	Group and List Management Server
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol (Secure)
IEC	International Engineering Consortium
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMAP	Internet Message Access Protocol
IMPP	Instant Messaging and Presence Protocol
IMPS	Instant Messaging and Presence Service
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPsec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ITU-T	International Telecommunications Union –Telecommunications Standard Committee

IRC	Internet Relay Chat
JPEG	Joint Photographic Experts Group
MGCF	Media Gateway Control Function
MIME	Multipurpose Internet Mail Extensions
MMS	Multimedia Messaging Service
MPEG	Moving Pictures Experts Group
MRFC	Multimedia Resource Function Controller
MRFP	Multimedia Resource Function Processor
MSRP	Message Session Relay Protocol
OMA	Open Mobile Alliance
PIDF	Presence Information Data Format
POP	Post Office Protocol
PRIM	Presence and Instant Messaging Protocol
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFC	Request For Comments
SAP	Service Access Point
SIMPLE	SIP for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
S/MIME	Secure MIME
SMS	Short Message Service
SSP	Server-Server Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UAC	User agent client
UAS	User agent server
UA	User agent
UDP	User Datagram Protocol
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
URI	Uniform Resource Identifier

VoIP	Voice over IP
WAP	Wireless Application Protocol
WBXML	Wireless Binary XML
WLAN	Wireless Local Area Network
WSP	Wireless Session Protocol
XCAP	XML Configuration Access Protocol
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

Chapter-1

INTRODUCTION

1.1 INTRODUCTION

The IP Multimedia Subsystem (IMS) standard defines a generic architecture for offering Voice over IP (VoIP) and multimedia services. It is an international, recognized standard, first specified by the Third Generation Partnership Project (3GPP/3GPP2) and now being embraced by other standards bodies including ETSI/TISPAN. The standard supports multiple access types –including GSM, WCDMA, CDMA2000, Wireline broadband access and WLAN.

For users, IMS-based services enable person-to-person and person-to-content communications in a variety of modes – including voice, text, pictures and video, or any combination of these – in a highly personalized and controlled way. IMS enables the efficient creation and delivery of an exciting range of emerging multimedia services that can be delivered over mobile, fixed, or converged mobile and fixed networks. Today's telephony and messaging services will be complemented by the next-generation of user-to-user applications.

As the Internet and mobile telecommunications becomes more and more integrated, new ways to communicate appears. Some of them, like WAP and MMS, have already made their way into every day life and others are on the way. One of the new upcoming applications is instant messaging. Concurrently with the instant messaging revolution of the Internet, the introduction of text messaging has launched a similar phenomenon in mobile networks. Now, instant messaging is about to make the transition from the wireline Internet to mobile networks.

This thesis studies the requirements placed upon instant messaging systems by mobile environments and a standard instant messaging protocol based on SIP, called SIMPLE, has been proposed and is under development. SIMPLE can also carry presence information, conveying a person's willingness and ability to engage in communications. Presence information is most recognizable today as buddy status in IM clients such as MSN Messenger and AIM.

1.2 OBJECTIVE OF THE WORK

With the growth in data traffic based on IP and the Internet explosion in the 1990s, next-generation network architectures were developed with aim of simplifying networks using a common platform to support multiple information streams.

The telecommunications industry is now evolving beyond VoIP in the wireline world to the next step in multimedia convergence. IMS is intended to serve as the next-generation converged platform that will transform the future of voice, video, and data communications for consumers and businesses. Essentially, IMS fills the gaps between all the disparate telecommunications technologies in the marketplace, including the gap between the public network and cellular voice, the gap between cellular voice and cellular data, and the gap between dedicated, purpose-built networks and open, standards-based networks. The filler in those gaps is SIP, the application protocol that is intended to bridge the myriad technologies in the marketplace and fuel today's commercial VoIP services.

The main aim of this thesis was to design two mobile clients and set up an IMS session between them and implementing instant messaging service using protocol instant messaging presence services (IMPS) and SIP for instant messaging and presence leveraging extensions (SIMPLE) and comparative study is done between on the basis of their functionalities.

1.3 ORGANISATION

This thesis starts by presenting the theoretical background needed to understand the basic structure of IP multimedia subsystems as described in chapter 2 and 3 and implementation of instant messaging as an application of IMS is described in chapter 4 and 5. A brief summary of all the chapters is given below:

Chapter 2: IP Multimedia Subsystems

In this chapter a brief introduction to IMS is discussed. Origin of IMS is discussed with its history. A different approach towards IMS is also described with its standards. Then the brief review of IMS application is discussed at the end of chapter.

Chapter 3: Basic architecture of IMS

In this chapter history of the circuit-switched and the packet-switched domains is described. In addition, introduction to design principles that lay behind the IMS architecture and its protocols is also explained. How to tackle the IMS network nodes and the different ways in which users are identified in the IMS are also described.

Chapter 4: Session Initiation Protocol

This chapter deals with description of SIP and explains how the basic SIP session is established. It also describes the structure of a SIP message.

Chapter 5: Implementation of Instant messaging and presence services

This chapter covers the new concepts of presence and instant communications, and shows how they can be implemented using IMPS and SIP. The basic protocol needed to implement instant messaging is IMPS & SIMPLE and architectural details are discussed.

Chapter 6: Results & Discussion

This chapter concludes the results with output logs generated during implementation of instant messaging services and presence. It also deals with comparison of both the protocols used to implement IMPS on the basis of their features

Chapter7: Conclusion & Future scope

In this chapter various features of instant messaging with presence are listed and the future utility of the thesis is discussed.

Chapter-2

IP MULTIMEDIA SUBSYSTEMS (IMS)

2.1 INTRODUCTION

Wireless Service providers are looking for new enhanced services as the success of particular services is difficult to predict, service providers need a dynamic architecture which will allow them to offer promising new services quickly, add resources for successful services as demand increases, and downplay or remove unsuccessful services easily. Such architecture is the IP Multimedia Subsystem (IMS), a modular standards-based service platform that uses the Internet Protocol (IP) and the Session Initiation Protocol (SIP).

2.2 ORIGIN OF IMS

IMS is an umbrella framework for providing enhanced IP-based services developed by the Third Generation Partnership Project (3GPP), collaboration among a number of telecommunications standards bodies. The original scope was to develop specifications for a 3rd Generation Mobile System based on evolved Global System for Mobile communication (GSM) core networks and the radio access technologies that they support. The scope was later amended to include the maintenance and development of the GSM specification and its related radio access technologies. To date, two “phases,” known as Release 5 and Release 6, have been published for IMS, and the specification details a framework for an IP/SIP-based network services architecture that is designed to span wireless, wireline, and cable networks.

2.2.1 History

- IMS was originally defined by the 3rd Generation Partnership Project (3GPP), as part of their standardization work for 3G mobile phone systems in UMTS networks. It first appeared in release 5 (evolution from 2G to 3G networks), when SIP-based multimedia was added. Support for the older GSM and GPRS networks was also provided.
- "Early IMS" was defined for IPv4 networks, and provided a migration path to IPv6.

- 3GPP2 (a different organization) based their CDMA2000 Multimedia Domain (MMD) on 3GPP IMS, adding support for CDMA2000
- In 3GPP release 6, interworking with WLAN was added.
- 3GPP release 7 added supports for fixed networks, by working together with TISPAN R1.

2.3 IMS: A DIFFERENT APPROACH

Unlike other approaches that are aimed at simply carrying circuit services on top of IP, the IMS framework allows operators to build an open IP-based service infrastructure that will enable easy deployment of rich multimedia communications services. IMS is intended to address the following network and user requirements:

• Deliver person-to-person real-time IP-based multimedia communications such as voice or video telephony as well as person-to-machine communications such as gaming, video-on-demand, and web surfing.

• Fully integrate real-time communications, such as live streaming and chat, with non-real-time multimedia.

• Enable multiple services and applications to interact, for example, video conferencing and gaming or real-time video and instant messaging combined with presence.

• Escalate communications sessions easily, for example, by turning an instant messaging session into a voice session with “one click.”

IMS has taken on increasing importance because network operators today need to converge traditional telecom services, such as voice calls and short message service (SMS) and data services, such as email, web browsing, and instant messaging (IM). Just as cable providers are exploring the possibility of delivering telecom services along with their video services, network operators need the ability to provide video services to remain competitive. In addition, customers are expecting converged services, and network operators now have an opportunity to deliver them with IMS.

2.4 STANDARDS FOR IMS

The Third Generation Partnership Project (3GPP)[2], focused on the needs of mobile operators, has led the major effort to standardize IMS. Other standards bodies have adopted most core functions of IMS for their respective domains and have

developed relevant segment-specific IMS extensions for next-generation mobile, wireline, and cable services (refer to Table 2.1).

Standards Organization or Consortium	Scope or Focus	Standards Contribution
Internet Engineering Task Force (IETF)	All IP networks	SIP and other protocols (e.g., COPS, Diameter, etc.)
Third Generation Partnership Project (3GPP)	Universal Mobile Telecommunications Service (UMTS) (Wideband Code Division Multiple Access [W-CDMA]) mobile networks and other access networks	IP Multimedia Subsystems (IMS)
Third Generation Partnership Project 2 (3GPP2)	CDMA2000 mobile networks and other access networks	Multimedia Domain (MMD)
European Telecom Standards Institute (ETSI)	Next –generation wireline networks	NGN effort by TISPAN NGN
International Telecommunication Union (ITU-T)	Next-generation wireline networks	Focus Group on Next Generation Networks (FGNGN) effort by ITU-T SG13 NGN and other ITU-T Study Groups
CDMA Development Group (CDG)	All mobile networks	Open Mobile Alliance (OMA) and Push-to-Talk over Cellular

Table 2.1: IMS Standards Organizations and Industry Forums

IMS has evolved from mobile wireless access (as per 3GPP R5) and is intended to be access-agnostic so that multiple access technologies can be blended—wireless fidelity (Wi-Fi), Worldwide Interoperability for Microwave Access (WiMAX), DSL, broadband cable access, and even enterprise level T1 (as per 3GPP R6 and R7). The IMS layered

approach decouples the network infrastructure from services with a standardized, horizontal approach (compared to traditional vertical silo service approaches).

The profitability appeal of IMS for service providers lies in its ability to provide a standard platform to respond rapidly to marketplace dynamics of revenue decline and the need to better address service personalization (for example, self-subscription, buddy lists, etc.) and control (for example, quality of service (QoS), class of service (CoS), charging, security, content filtering, etc.). Essentially, IMS is an application-centric concept appealing to all types of service providers. Figure 2.1 provides a simple representation of IMS in three planes—the middle IMS layer separates the services or application layer from the transport layer for greater flexibility.

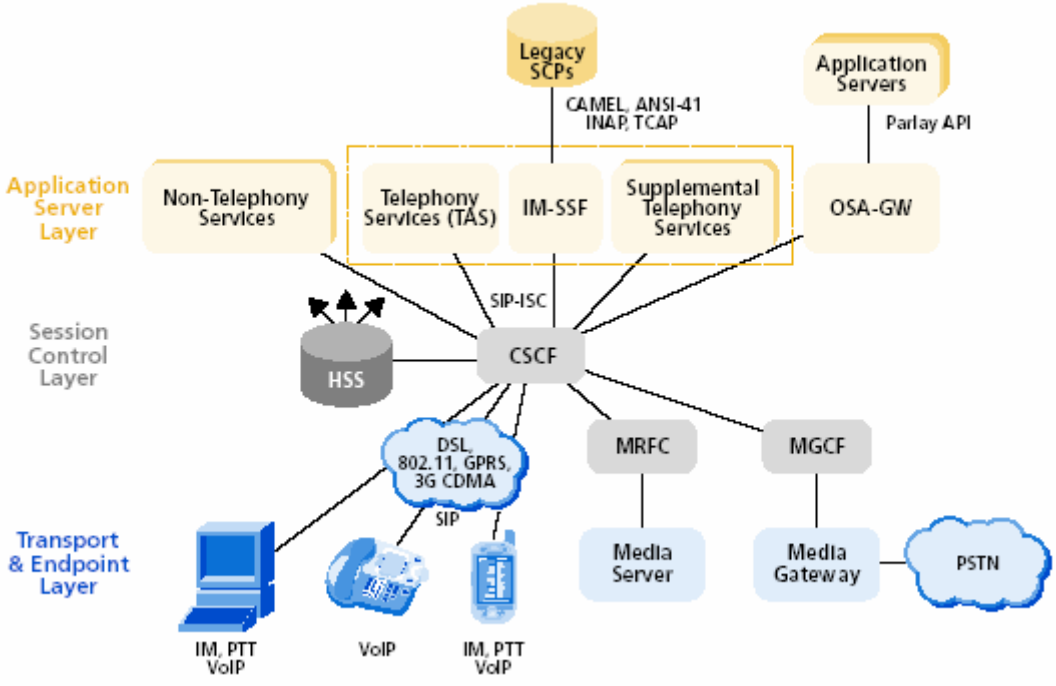

Figure 2.1 Simplified view of layered architecture in IMS

2.5 IMS APPLICATIONS

IMS enables the efficient creation and delivery of an exciting range of emerging multimedia services that can be delivered over mobile, fixed, or converged mobile and fixed networks. It introduces a multimedia call model that enables consistency in the user experience, accelerated service development, and the efficient and flexible delivery of rich multimedia content and services. Today's telephony and messaging services will be

complemented by the next-generation of user-to-user applications. This will make collaboration faster and easier, because subscribers will be able to share everything from documents and whiteboards to gaming experiences. Table 2.2[1] shows a few examples of multimedia services that will be enabled by IMS.

Instant Messaging	Existing data services can be enhanced to contain any MIME-type media content. Users can communicate in real time using not only text, but also images, audio clips, video snips, and document sharing.
Push-to-Talk	Providers can lower costs through half-duplex telephony services using a model already established for radio services.
Shared Documents	Users can access, review, and edit a single document—with each user seeing all the changes in real time. They can discuss the changes via phone or by using instant messaging to streamline review cycles and increase productivity
Video Conferencing	Traditional one-to-one video conferencing services can be extended. Each location establishes a videoconference to a central bridge, which links the calls and applies the appropriate service logic to meet QoS requirements
Voice Mail	Voice messages can be converted into audio files and distributed to roaming subscribers via instant messaging. Services can even be offered to support video voice mail.
VoIP	Established carriers can convert circuit-switched voice calls transparently onto IP infrastructure, and carriers with IP infrastructure in place can provide enhanced VoIP and multimedia services to a broad range of communications devices.
Web-Based Conferencing	Creating voice or multimedia conferences can be as easy as clicking to invite attendees and establishing the session via your Web browser.

Table 2.2: IMS Applications

These are just a sampling of the exciting multimedia services that will be delivered using IMS infrastructure. Service providers also will be able to create innovative new offerings that bridge mobile and fixed networks, such as content sharing or messaging services between a personal digital assistant (PDA) and a PC.

Chapter-3

BASIC ARCHITECTURE OF IMS

3.1 INTRODUCTION

The IP-Multimedia Subsystem (IMS) defines the functional architecture for a managed IP-based network. It aims to provide a means for carriers to create an open, standards-based network that delivers integrated multimedia services to increase revenue, while also reducing network capital expenditure (CapEx) and operational expenditure (OpEx). IMS was originally designed for third-generation mobile phones, but it has already been extended to handle access from WiFi networks, and is continuing to be extended into an access-independent platform for service delivery, including broadband fixed-line access. It promises to provide seamless roaming between mobile, public WiFi and private networks for a wide range of services and devices.

The IMS architecture has been designed to enable operators to provide a wide range of real-time, packet-based services and to track their use in a way that allows both traditional time-based charging as well as packet and service-based charging. It has become increasingly popular both with wireline and wireless service providers as it is designed to increase carrier revenues, deliver integrated multimedia services, and create an open, standards-based network.The IP Multimedia Subsystem (IMS) is a standardized Next Generation Networking (NGN) architecture for telecom operators that want to provide mobile and fixed multimedia services. It uses a Voice-over-IP (VoIP) implementation based on a 3GPP standardized implementation of SIP, and runs over the standard Internet Protocol (IP). Existing phone systems (both packet-switched and circuit-switched) are supported.

3.2 BASIC PRINCIPLE

Access independence: IMS will eventually work with any network (fixed, mobile or wireless) with packet-switching functions, such as GPRS, UMTS, CDMA2000,

WLAN, WiMAX, DSL, cable, ... Older circuit-switched phone systems (POTS, GSM) are supported through gateways. Open interfaces between control and service layers allow elements and calls/sessions from different access networks to be mixed.

Different network architectures: IMS allows operators and service providers to use different underlying network architectures.

Terminal and user mobility: The mobile network provides terminal mobility (roaming), while user mobility is provided by IMS and SIP

Extensive IP-based services: IMS should make it easier to offer just about any IP-based service. Examples include voice over IP (VOIP), Push to talk over cellular (POC), videoconferencing, instant messaging, presence information .

3.3 ARCHITECTURE OF IMS

The IP Multimedia Core Network Subsystem is a collection of different functions, linked by standardized interfaces. A function is not a node (hardware box): an implementer is free to combine 2 functions in 1 node, or to split a single function into 2 or more nodes. IMS distributes much of the intelligence to the communications device or the edge of the network, allowing carriers to develop multimedia services that can be delivered and managed across diverse access networks. Because service intelligence is largely distributed to the edge of the network or to the communications device, network operators can more swiftly create enhanced services that can be provisioned across multiple networks. IMS is a strategic technology for next-generation services, and it is offers a standards-based architecture for critical functions such as:

1) Call control

2) Presence

3) Location

4) Content-based billing

5) Profile management

6) Convergence

7) Service interaction

8) Abstract data management and distribution

Network operators create a single IP, asynchronous transfer mode (ATM), or multi-protocol label switching (MPLS) core network for transport, and they can implement IMS architecture across mobile and/or fixed networks. Subscribers can be provided with flexible means of accessing services delivered over IMS infrastructure. They can access IMS services by dialing up over the PSTN, or they can benefit from more rich multimedia services by accessing the infrastructure through the PSTN using digital subscriber line (DSL) services. They can also access IMS services via broadband cable technology, and mobile users can reap the benefits of IMS via cell phones or WiFi connections.

In this converged network architecture, IMS provides an underlying infrastructure mechanism for the intelligent interaction of applications and services. Common features and capabilities can be reused across many applications in a "develop once, use many" fashion. For example, presence information can be re-used for applications ranging from push-to-talk to multimedia conferencing. This flexible, open architecture enables service brokering to work in an intelligent way to support feature interaction across multiple applications.

3.4 NETWORK ELEMENTS OF IMS

The IMS architecture defines the logical elements necessary to implement next-generation multimedia services across multiple network types. It is important to note that these logical functions do not necessarily have a one-to-one relationship with physical equipment. The components of the IMS architecture refer to functions, not platforms. Multiple functions can be mapped to a single network device, and, conversely, a single function can conceivably be implemented across multiple physical platforms.

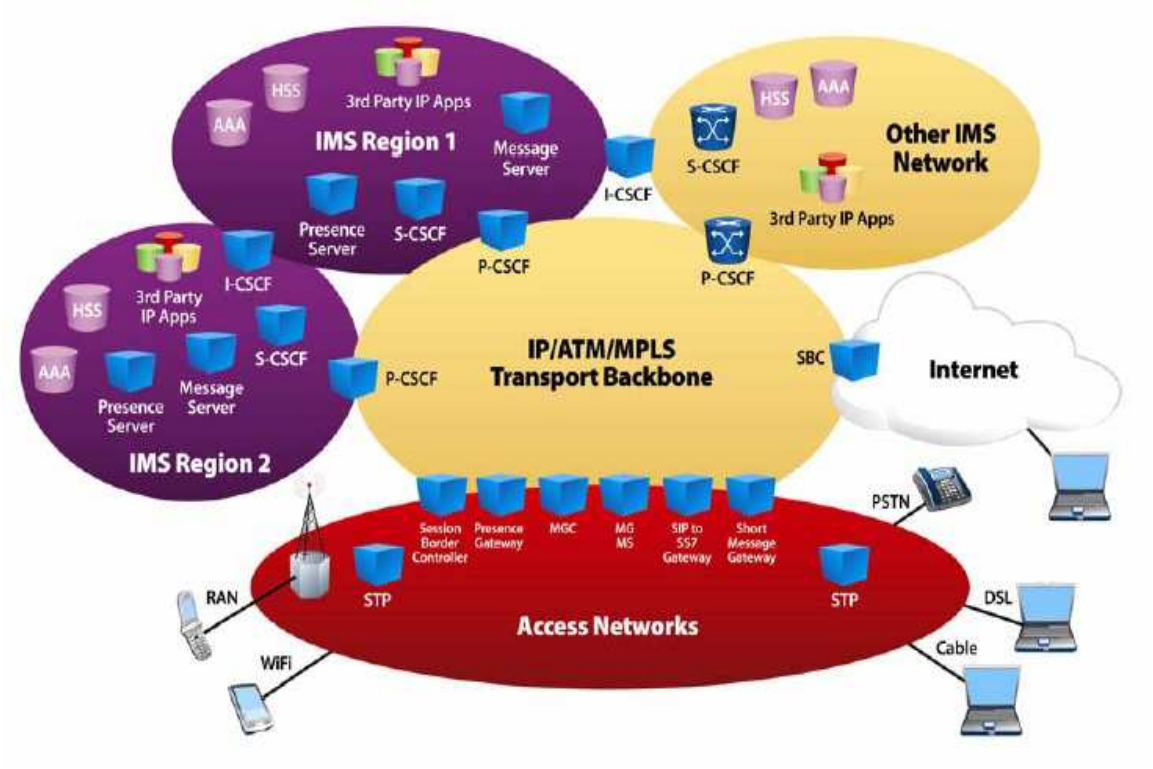

Figure 3.1 Architecture of IMS[1]

The Figure 3.1 shows the nodes included in the so-called IP Multimedia Core Network Subsystem. These nodes are:

1. One or more user databases, called HSSs (Home Subscriber Servers) and SLFs (Subscriber Location Functions).
2. One or more SIP servers, collectively known as CSCFs (Call/Session Control Functions).
3. One or more ASs (Application Servers).
4. One or more MRFs (Media Resource Functions), each one further divided into MRFC (Media Resource Function Controllers) and MRFP (Media Resource Function Processor).
5. One or more BGCFs (Breakout Gateway Control Functions).
6. One or more PSTN gateways, each one decomposed into an SGW (Signaling Gateway), an MGCF (Media Gateway Controller Function), and an MGW (Media Gateway).The following are descriptions of elements and concepts of IMS which are shown in figure3.1.

3.4.1The Databases: The HSS and The SLF:

The home subscriber service (HSS) manages subscriber information and enables users or servers to locate targets. The profile and the preferences of each user are stored in the HSS database. By centralizing this information, service providers can simplify administration and ensure a consistent view of active subscribers across all services. With IMS, managing mobility is easier than ever. The HSS contains the subscriber information and allows subscribers to locate each other. The Home Subscriber Server (HSS) is the central repository for user-related information. Technically, the HSS is an evolution of the HLR (Home Location Register), which is a GSM node. The HSS contains all the user-related subscription data required to handle multimedia sessions. These data include, among other items, location information, security information (including both authentication and authorization information), user profile information (including the services that the user is subscribed to), and the S-CSCF (Serving-CSCF) allocated to the user.

The SLF is a simple database that maps users' addresses to HSSs. A node that queries the SLF, with a user's address as the input, obtains the HSS that contains all the information related to that user as the output. It maps user addresses for a Packet and Public Land Mobile Network (PLMN), and stores information such as the type of phone each customer is using and their location when they attempt to use a service.

A network may contain more than one HSS, in case the number of subscribers is too high to be handled by a single HSS. In any case, all the data related to a particular user are stored in a single HSS. Networks with a single HSS do not need an SLF. On the other hand, networks with more than one HSS do require an SLF(Subscriber Location Function) .

3.4.2 Call Session Control Function

The Call Session Control Function consists of several "sub-elements" that handle all of the signaling associated with call setup and teardown and basic SIP message exchange. The CSCF can also handle signaling for control of the IP media itself and the session initiation stack is within its domain. The sub-elements are collectively known as CSCFs, but any CSCF belongs to one of the following three categories.

- P-CSCF (Proxy-CSCF).

- I-CSCF (Interrogating-CSCF).
- S-CSCF (Serving-CSCF).

3.4.2.1Proxy-Call Session Control Function

The Proxy-CSCF is the exclusive point of contact for user equipment. All traffic from the end user comes into the IMS through the P-CSCF, which performs the following functions:

• Handles initial security by querying the HSS

• Verifies SIP messages

• Performs IPSec integrity protection to create trusted messages

• Compresses messages to reduce latency

• Creates billing information

Today, the P-CSCF is typically located in the home network but will perhaps be moved to the visited network in the future.

The P-CSCF is the first point of contact (in the signaling plane) between the IMS terminal and the IMS network. From the SIP point of view the P-CSCF is acting as an outbound/inbound SIP proxy server. This means that all the requests initiated by the IMS terminal or destined to the IMS terminal traverse the P-CSCF. The P-CSCF forwards SIP requests and responses in the appropriate direction (i.e., toward the IMS terminal or toward the IMS network).The P-CSCF is allocated to the IMS terminal during IMS registration and does not change for the duration of the registration (i.e., the IMS terminal communicates with a single P-CSCF during the registration).

The P-CSCF includes several functions, some of which are related to security. First, it establishes a number of IPsec security associations toward the IMS terminal. These IPsec security associations offer integrity protection (i.e., the ability to detect that the contents of the message have changed since its creation).

Once the P-CSCF authenticates the user (as part of security association establishment) the P-CSCF asserts the identity of the user to the rest of the nodes in the network. This way, other nodes do not need to further authenticate the user, because they trust the P-CSCF. The rest of the nodes in the network user's identity (asserted by the P-

CSCF) have a number of purposes, such as providing personalized services and generating account records. Additionally, the P-CSCF verifies the correctness of SIP requests sent by the IMS terminal. This verification keeps IMS terminals from creating SIP requests that are not built according to SIP rules.

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

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

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

3.4.2.2 P-CSCF Location

The P-CSCF may be located either in the visited network or in the home network. In case the underlying packet network is based on GPRS, the P-CSCF is always located in the same network where the GGSN (Gateway GPRS Support Node) is located. So both P-CSCF and GGSN are either located in the visited network or in the home network. Due to current deployments of GPRS, it is expected that the first IMS networks will inherit this mode and will be configured with the GGSN and P-CSCF in the home network. It is also expected that once IMS reaches the mass market,

operators will migrate the configuration and will locate the P-CSCF and the GGSN in the visited network.

3.4.2.3 Serving Call Session Control Function

The Serving-CSCF handles all of the SIP signaling that goes on between endpoints and provides the following functions:

• Provides SIP routing by translating public user identities (today, phone numbers but may change in the future) to endpoint IP addresses and sending messages to the application servers

• Holds session context, which is a binding between user locations (for example, the IP address of user equipment) and the SIP address of record, which is the public user identity

• Performs session control

• Prevents the unauthorized use of services

The serving-CSCF (S-CSCF) performs the session control services for subscribers. It maintains session state as needed by the network operator for support of the services and is the core session control function for IMS. It maintains session state for each current user and enables communications with servers of applications and content. The S-CSCF manages all session control messages, and it sends information to the users involved in a session, such as alerts to conference callers about an attendee entering or leaving the session. The S-CSCF is the central node of the signaling plane. The S-CSCF is essentially a SIP server, but it performs session control as well. In addition to SIP server functionality the S-CSCF also acts as a SIP registrar. This means that it maintains a binding between the user location (e.g., the IP address of the terminal the user is logged on) and the user's SIP address of record (also known as a Public User Identity).

The main reasons to interface the HSS are:

- To download the authentication vectors of the user who is trying to access the IMS from the HSS. The S-CSCF uses these vectors to authenticate the user.

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

All the SIP signaling the IMS terminals sends, and all the SIP signaling the IMS terminal receives, traverses the allocated S-CSCF. The S-CSCF inspects every SIP message and determines whether the SIP signaling should visit one or more application servers en route toward the final destination. Those application servers would potentially provide a service to the user.

One of the main functions of the S-CSCF is to provide SIP routing services. If the user dials a telephone number instead of a SIP URI the S-CSCF provides translation services. The S-CSCF also enforces the policy of the network operator. For example, a user may not be authorized to establish certain types of sessions. The S-CSCF keeps users from performing unauthorized operations. A network usually includes a number of S-CSCFs for the sake of scalability and redundancy. Each S-CSCF serves a number of IMS terminals, depending on the capacity of the node.

The S-CSCF is always located in the home network.

3.4.2.4 Interrogating Call Session Control Function

The Interrogating-CSCF is the first point of contact within the home network, and the I-CSCF contacts the HSS to locate the Serving-CSCF for a particular subscriber. The I-CSCF may provide a Topology Hiding Inter-network Gateway (THIG), which helps to protect the IMS network topology by encrypting some of the internal IP addresses and other network information within the SIP messages.

The I-CSCF is a SIP proxy located at the edge of an administrative domain. The address of the I-CSCF is listed in the DNS (Domain Name System) records of the domain. When a SIP server follows SIP procedures to find the next SIP hop for a particular message the SIP server obtains the address of an I-CSCF of the destination domain. Besides the SIP proxy server functionality the I-CSCF has an interface to the SLF and the HSS. The I-CSCF retrieves user location information and routes the SIP request to the appropriate destination (typically an S-CSCF).The I-CSCF is responsible

for topology hiding to prevent foreign networks from gaining visibility into a network operator's infrastructure. It identifies which S-CSCF will process SIP requests for a given user, and it leverages information from the HSS to forward all session-related messages to the right S-CSCF.

A network will include typically a number of I-CSCFs for the sake of scalability and redundancy. The I-CSCF is usually located in the home network, although in some especial cases, such as an I-CSCF(THIG), it may be located in a visited network as well.

3.4.3 Application Server

Application servers provide multimedia services to the IMS network by providing access to all other IMS elements as SIP servlets. New services are also deployed through the application servers, and since IMS has a modular architecture, only the application servers need to be replaced or upgraded when new services are deployed. Such a strategy is in direct contrast to previous vertical models in which services were deployed as point solutions, each with its own set of proprietary equipment. The application servers can be located either in the home network or in a third-party network, and in the latter case, these application servers do not interface with the HSS. Since the applications servers are OSA applications servers, they can access IMS securely from external networks and interface to GSM CAMEL servers. Figure 3.2 depicts three different types of Application Servers:

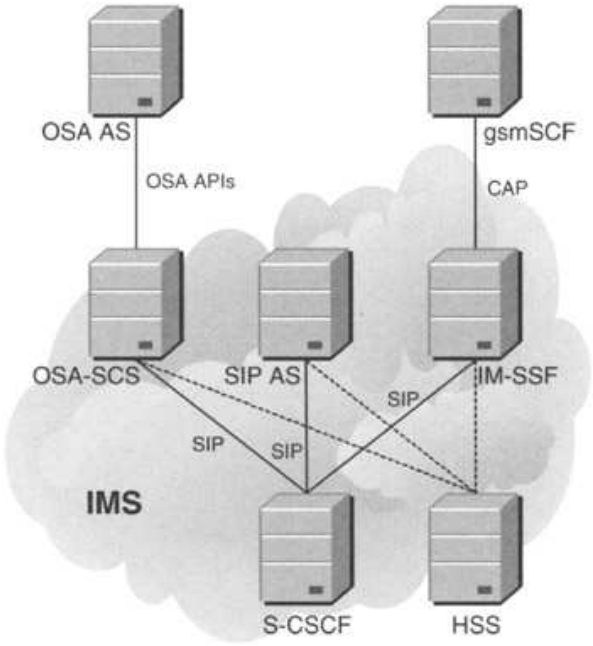

Figure 3.2: Three types of Application Servers[2]

- **SIP AS (Application Server):** This is the native Application Server that hosts and executes IP Multimedia Services based on SIP. It is expected that new IMS-specific services will likely be developed in SIP Application Servers.

- **OSA-SCS (Open Service Access—Service Capability Server):** This application server provides an interface to the OSA framework Application Server. It inherits all the OSA capabilities, especially the capability to access the IMS securely from external networks. This node acts as an Application Server on one side (interfacing the S-CSCF with SIP) and as an interface between the OSA Application Server and the OSA Application Programming Interface

- **IM-SSF (IP Multimedia Service Switching Function):** This specialized application server allows us to reuse CAMEL (Customized Applications for Mobile network Enhanced Logic) services that were developed for GSM in the IMS. The IM-SSF allows a gsmSCF (GSM Service Control Function) to control an IMS session. The IM-SSF acts as an Application Server on one side (interfacing the S-CSCF with SIP). On the other side, it acts as an SCF (Service Switching Function), interfacing the gsmSCF with a protocol based on CAP.

All three types of application servers behave as SIP application servers toward the IMS network (i.e., they act as either a SIP proxy server, a SIP User Agent, a SIP redirect server or a SIP Back-to-back User Agent). The IM-SSF AS and the OSA-SCS AS have other roles when interfacing CAMEL or OSA, respectively.

In addition to the SIP interface the AS may optionally provide an interface to the HSS. The SIP-AS and OSA-SCS interfaces toward the HSS are used to download or upload data related to a user stored in the HSS. The IM-SSF interface toward the HSS is based on MAP (Mobile Application Part).

3.4.4 Media Resources

An MRF (*Media Resource Function*) provides a source of media in the home network. It's used for:

Playing of announcements (audio/video)

Multimedia conferencing (e.g. mixing of audio streams)

Text-to-speech conversation (TTS) and speech recognition.

Real-time transcoding of multimedia data (i.e. conversion between different codecs)

Each MRF is further divided into:

An MRFC (Media Resource Function Controller) is a signaling plane node that acts as a SIP User Agent to the S-CSCF, and which controls the MRFP with a H.248 interface

An MRFP (Media Resource Function Processor) is a media plane node that implements all media-related functions.

The MRF is always located in the home network.

3.4.5 Gateway Control Functions

The gateway control functions manage media gateways (MGs) and handle the communications between the IP and SS7 networks to enable interworking with the PSTN. The breakout gateway control function (BGCF) selects the network in which the connection to the PSTN is to occur for a given session. If the BGCF determines that the breakout is to occur in the same network in which the BGCF is located, then the BGCF will select a media gateway control function (MGCF) element, which will be responsible for the interworking with the PSTN for signaling. If the breakout is in another network, the BGCF will forward this session signaling to another BGCF, or an MGCF, depending on the configuration, in the selected network.

3.4.5.1 The BGCF

The BGCF is essentially a SIP server that includes routing functionality based on telephone numbers. The BGCF is only used in sessions that are initiated by an IMS terminal and addressed to a user in a circuit-switched network, such as the PSTN or the PLMN. The main functionality of the BGCF is:

a) To select an appropriate network where interworking with the circuit-switched domain is to occur.

b) Or, to select an appropriate PSTN/CS gateway, if interworking is to occur in the same network where the BGCF is located.

3.4.5.2 The PSTN/CS Gateway

The PSTN gateway provides an interface toward a circuit-switched network, allowing IMS terminals to make and receive calls to and from the PSTN (or any other circuit-switched network).Figure 3.3 shows a BGCF and a decomposed PSTN gateway that interfaces the PSTN. The PSTN gateway is decomposed into the following functions.

- **SGW (Signaling Gateway):** the Signaling Gateway interfaces the signaling plane of the CS network (e.g., the PSTN). The SGW performs lower layer protocol conversion. For instance, an SGW is responsible for replacing the lower MTP transport with SCTP (Stream Control Transmission Protocol) over IP.

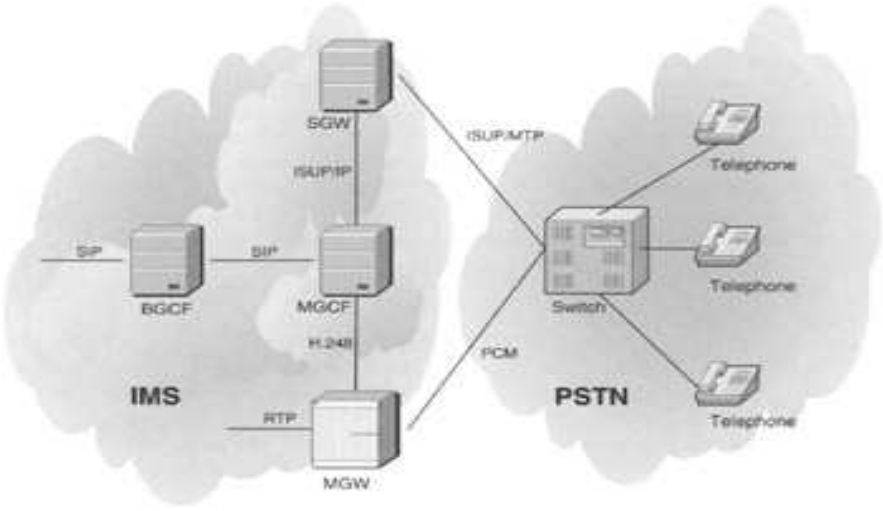

Figure 3.3 The PSTN/CS gateway interfacing a CS network

- **MGCF (Media Gateway Control Function):** the MGCF is the central node of the PSTN/CS gateway. It implements a state machine that does protocol conversion and maps SIP (the call control protocol on the IMS side) to either ISUP over IP or BICC over IP (both BICC and ISUP are call control protocols in circuit-switched networks). In addition to the call control protocol conversion the MGCF controls the resources in an MGW (Media Gateway).

- **MGW (Media Gateway):** the Media Gateway interfaces the media plane of the PSTN or CS network. On one side the MGW is able to send and receive IMS media over the Real-Time Protocol (RTP). On the other side the MGW uses one or more PCM (Pulse Code Modulation) time slots to connect to the CS network. Additionally, the MGW performs transcoding when the IMS terminal does not support the codec used by the CS side.

3.5 HOME NETWORKS AND VISITED NETWORKS

The IMS borrows a few concepts from GSM and GPRS, such as having a home and a visited network. In the cellular model, when we use our cell phones in the area where we reside, we are using the infrastructure provided by our network operator. This infrastructure forms the so-called home network. On the other hand, if we roam outside the area of coverage of our home network (e.g., when we visit another country), we use an infrastructure provided not by our operator, but by another operator. This infrastructure is what we call the visited network, because effectively we are a visitor in this network. In order to use a visited network the visited network operator has to have signed a roaming agreement with our home network operator. In these agreements both operators negotiate some aspects of the service provided to the user, such as price of calls, quality of service, or how to exchange accounting records.

The IMS reuses the same concept of having a visited and a home network. Most of the IMS nodes are located in the home network, but there is a node that can be either located in the home or the visited network. That node is the P-CSCF (Proxy-CSCF). The IMS allows two different configurations, depending on whether the P-CSCF is located in the home or visited network. Additionally, when the IP-CAN (IP Connectivity Access Network) is GPRS the location of the P-CSCF is subordinated to the location of the GGSN. In roaming scenarios, GPRS allows location of the GGSN either in the home or in the visited network (the SGSN is always located in the visited network).

In the IMS, both the GGSN and the P-CSCF share the same network. This allows the P-CSCF to control the GGSN over the so-called Go interface. As both the P-CSCF and the GGSN are located in the same network the Go interface is always an intra-operator interface, which makes its operation simpler.

Figure3.4 shows a configuration where the P-CSCF (and the GGSN) is located in the visited network. This configuration represents a longer term vision of the IMS, because it requires IMS support from the visited network .It is not expected that all networks in the world will deploy IMS simultaneously.

Consequently, it is not expected that all roaming partners will upgrade their GGSNs at the same time the home network operator starts to provide the IMS service. So, we expect that early IMS deployments will locate the P-CSCF in the home network, as shown in Figure 3.5.

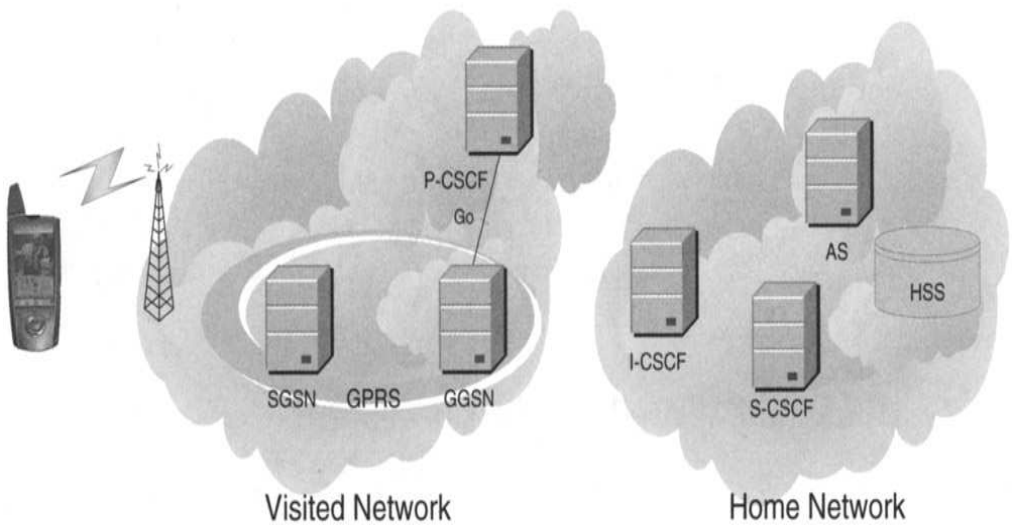

Figure 3.4 The P-CSCF located in the visited network

This figure shows a near term configuration where both the P-CSCF and the GGSN are located in the home network. This configuration does not require any IMS support from the visited network. Particularly, the visited network does not need to have a 3GPP Release 5 compliant GGSN. The visited network only provides the radio bearers and the SGSN. So, this configuration can be deployed from the very first day of the IMS. As a consequence, it is expected that this will be the most common configuration in the early years of IMS deployments. Even so, this configuration has a severe disadvantage with respect to the configuration where the P-CSCF and GGSN are located in the visited network. Since the media plane traverses the GGSN and the GGSN is located in the home network the media are first routed to the home network and then to their destination.

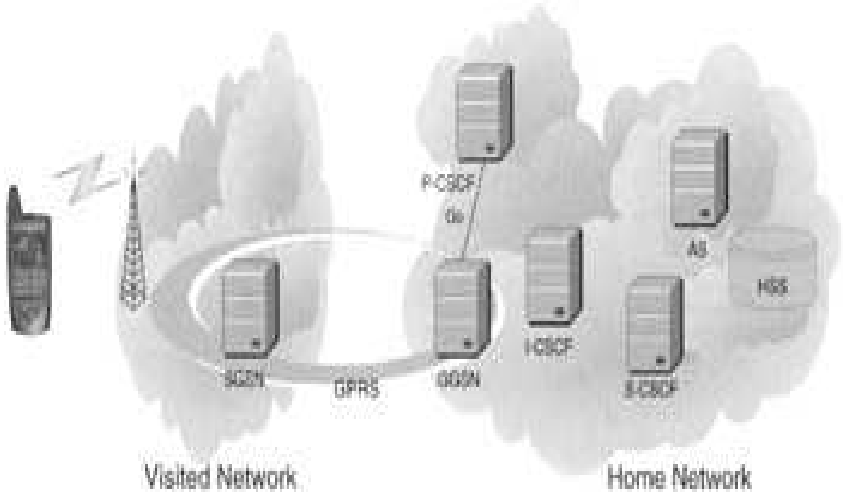

Figure 3.5 The P-CSCF located in the home network

This creates an undesired trombone effect that causes delays in the media plane.

There is a misconception that compression between the IMS terminal and the P-CSCF is enabled just to save a few bytes over the air interface. This is not the motivation lying behind compression. Particularly, it is not worth saving a few bytes of signaling when the IMS terminal will be establishing a multimedia session (e.g., audio, video) that will use much more bandwidth than the signaling. The main motivation for compression is to reduce the time to transmit SIP messages over the air interface.

3.5 IDENTIFICATION IN IMS

In a network of any kind, it must be possible to uniquely identify users. This is the property that allows a particular phone to ring (as opposed to a different telephone) when we dial a sequence of digits in the PSTN (Public Switched Telephone Network).Central to any network is the ability of the operator to identify users, so that calls can be directed to the appropriate user. In the PSTN, users are identified by a telephone number (i.e., a collection of ordered digits that identify the telephone subscriber). The telephone number that identifies a subscriber may be represented in different formats: a local short number, a long-distance number, or an international number.

In essence, these are just different representations of the same telephone subscriber. The length of the digits depends on the destination of the call (e.g., same area, another region, or another country).Additionally, when a service is provided; sometimes

there is a need to identify the service. In the PSTN services are identified by special numbers, typically through a special prefix, such as 800 numbers. IMS also provides mechanisms to identify services.

3.6.1 Public User Identities

In the IMS there is also a deterministic way to identify users. An IMS user is allocated with one or more *Public User Identities.* The home operator is responsible for allocating these Public User Identities to each IMS subscriber. A Public User Identity is either a SIP URI or a TEL URL. Public User Identities are used as contact information on business cards. In the IMS, Public User Identities are used to route SIP signaling. If we compare the IMS with GSM, a Public User Identity is to the IMS what an MSISDN (Mobile Subscriber ISDN Number) is to GSM.

When the Public User Identity contains a SIP URI, it typically takes the form of sip:first.last@operator.com, although IMS operators are able to change this scheme and address their own needs. Additionally, it is possible to include a telephone number in a SIP URI using the following format:

sip:+1–212–555–0293@operator.com;user=phone

This format is needed because SIP requires that the URI under registration be a SIP URI. So, it is not possible to register a TEL URL in SIP, although it is possible to register a SIP URI that contains a telephone number. The TEL URL is the other format that a Public User Identity can take. The following is a TEL URL representing a phone number in international format:

tel:+1–212–555–0293

TEL URLs are needed to make a call from an IMS terminal to a PSTN phone, because PSTN numbers are represented only by digits. On the other hand, TEL URLs are also needed if a PSTN subscriber wants to make a call to an IMS user, because a PSTN user can only dial digits.

We envision that operators will allocate at least one SIP URI and one TEL URL per user. There are reasons for allocating more than one Public User Identity to a user,

such as having the ability to differentiate personal (e.g., private) identities, that are known to friends and family from business Public User Identities (that are known to colleagues), or for triggering a different set of services.

The IMS brings an interesting concept: *a set of implicitly registered public user identities.* In regular SIP operation, each identity that needs to be registered requires a SIP REGISTER request. In the IMS, it is possible to register several Public User Identities in one message, saving time and bandwidth.

3.6.2 Private User Identities

Each IMS subscriber is assigned a *Private User Identity.* Unlike Public User Identities, Private User Identities are not SIP URIs or TEL URLs; instead, they take the format of a NAI (Network Access Identifier).

The format of a NAI is username@operator.com.

Unlike Public User Identities, Private User Identities are not used for routing SIP requests; instead, they are exclusively used for subscription identification and authentication purposes. A Private User Identity performs a similar function in the IMS as an IMSI (International Mobile Subscriber Identifier) does in GSM. A Private User Identity need not be known by the user, because it might be stored in a smart card, in the same way that an IMSI is stored in a SIM (Subscriber Identity Module).

3.6.3 The Relation between Public and Private User Identities

Operators assign one or more Public User Identities and a Private User Identity to each user. In the case of GSM/UMTS the smart card stores the Private User Identity and at least one Public User Identity. The HSS, as a general database for all the data related to a subscriber, stores the Private User Identity and the collection of Public User Identities allocated to the user. The HSS and the S-CSCF also correlate the Public and Private User Identities.

The relation between an IMS subscriber, the Private User Identity and the Public User Identities is shown in Figure 3.6. An IMS subscriber is assigned one Private User

Identity and a number of Public User Identities. This is the case of the IMS as standardized in 3GPP Release 5.

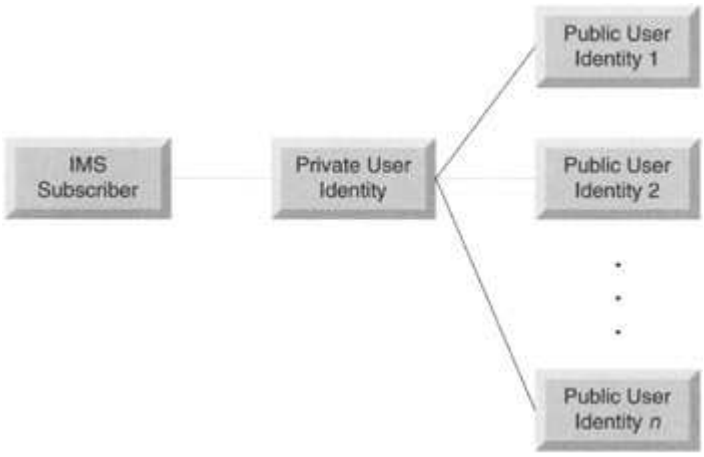

Figure 3.6: Relation of Private and Public User Identities in 3GPP R5

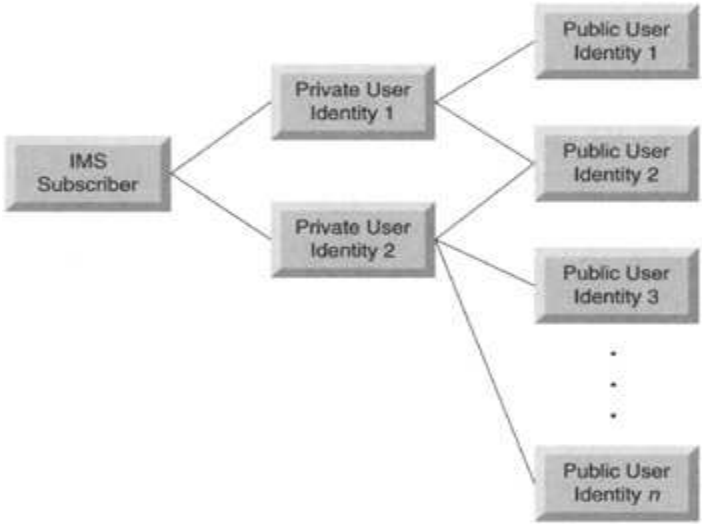

Figure 3.7 Relation of Private and Public User Identities in 3GPP R6

3GPP Release 6 has extended the relationship of Private and Public User Identities, as shown in Figure 3.7. An IMS subscriber is allocated not with one, but with a number of Private User Identities. In the case of UMTS, only one Private User Identity is stored in the smart card, but users may have different smart cards that they insert in different IMS terminals. It might be possible that some of those Public User Identities are used in combination with more than a single Private User Identity.

This is the case of Public User Identity #2 in Figure 3.7, because it is assigned to both Private User Identity #1 and #2.This allows Public User Identity #2 to be used simultaneously from two IMS terminals, each one assigned with a different Private User Identity (e.g., different smart cards are inserted in different terminals).

3.7 SIM, USIM, AND ISIM IN 3GPP

Central to the design of 3GPP terminals is the presence of a UICC (Universal Integrated Circuit Card). The UICC is a removable smart card that contains a limited storage of data. The UICC is used to store, among other things, subscription information, authentication keys, a phonebook, and messages.

GSM and 3GPP specifications rely on the presence of a UICC in the terminal for its operation. Without a UICC present in the terminal the user can only make emergency calls. The UICC allows users to easily move their user subscriptions (including the phonebook) from one terminal to another. The user simply removes the smart card from a terminal and inserts it into another terminal.UICC is a generic term that defines the physical characteristics of the smart card (like the number and disposition of pins, voltage values, etc.). The interface between the UICC and the terminal is standardized.

A UICC may contain several logical applications, such as a SIM (Subscriber Identity Module), a USIM (Universal Subscriber Identity Module), and an ISIM (IP multimedia Services Identity Module). Additionally, a UICC can contain other applications, such as a telephone book. Figure 3.8 represents a UICC that contains several applications.

3.7.1 SIM

SIM provides storage for a collection of parameters (e.g., user subscription information, user preferences, authentication keys, and storage of messages) that are essential for the operation of terminals in GSM networks. Although the terms UICC and SIM are often interchanged, UICC refers to the physical card, whereas SIM refers to a single application residing in the UICC that collects GSM user subscription information. SIM is widely used in 2G (Second Generation) networks, such as GSM networks.

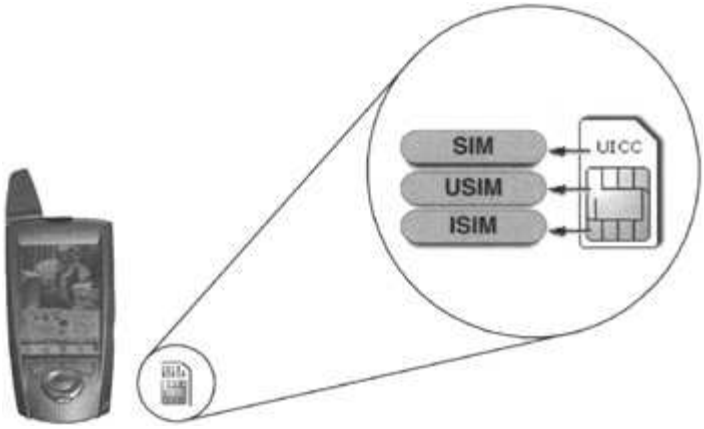

Figure 3.8 SIM, USIM, and ISIM in the UICC of 3GPP IMS terminals

3.7.2 USIM

USIM (standardized in 3GPP) is another example of an application that resides in third-generation UICCs. USIM provides another set of parameters (similar in nature, but different from those provided by SIM) which include user subscriber information, authentication information, payment methods, and storage for messages. USIM is used to access UMTS (Universal Mobile Telecommunication) networks, the third-generation evolution of GSM.

A USIM is required if a circuit-switched or packet-switched terminal needs to operate in a 3G (Third Generation) network. Obviously, both SIM and USIM can co-exist in the same UICC, so that if the terminal is capable, it can use both GSM and UMTS networks. Figure 3.9 shows a simplified version of the structure of USIM. USIM stores, among others, the following parameters:

- ✓ **IMSI (International Mobile Subscriber Identity):** IMSI is an identity assigned to each user. This identity is not visible to users themselves, but only to the network. IMSI is used as the user identification for authentication purposes. The Private User Identity is the equivalent of the IMSI in IMS.

- ✓ **MSISDN (Mobile Subscriber ISDN Number):** this field stores one or more telephone numbers allocated to the user. A Public User Identity is the equivalent of the MSISDN in the IMS.

- ✓ **CK (Ciphering Key) and IK (Integrity Key):** these are the keys used for ciphering and integrity protection of data over the air interface. USIM separately stores the keys used in circuit-switched and packet-switched networks.

- ✓ **Long term secret:** USIM stores a long term secret that is used for authentication purposes and for calculating the integrity and cipher keys used between the terminal and the network.

- ✓ **SMS (Short Messages Service):** USIM provides storage for short messages and their associated data (e.g., sender, receiver, and status).

- ✓ **SMS (Short Message Service) parameters:** this field in the USIM stores configuration data related to the SMS service, such as the address of the SMS centre or the protocols that are supported.

- ✓ **MMS (Multimedia Messaging Service) user connectivity parameters:** this field stores configuration data related to the MMS service, such as the address of the MMS server and the address of the MMS gateway.

- ✓ **MMS user preferences:** this field stores the user preferences related to the MMS service, such as the delivery report flag, read-reply preference, priority, and time of expiration.

3.7.3 ISIM

A third application that may be present in the UICC is ISIM (standardized in 3GPP). ISIM is of especial importance for the IMS, because it contains the collection of parameters that are used for user identification, user authentication, and terminal configuration when the terminal operates in the IMS. ISIM can co-exist with a SIM, a USIM, or both applications in the same UICC. Figure 3.10 depicts the structure of the ISIM application. The relevant parameters stored in ISIM are:

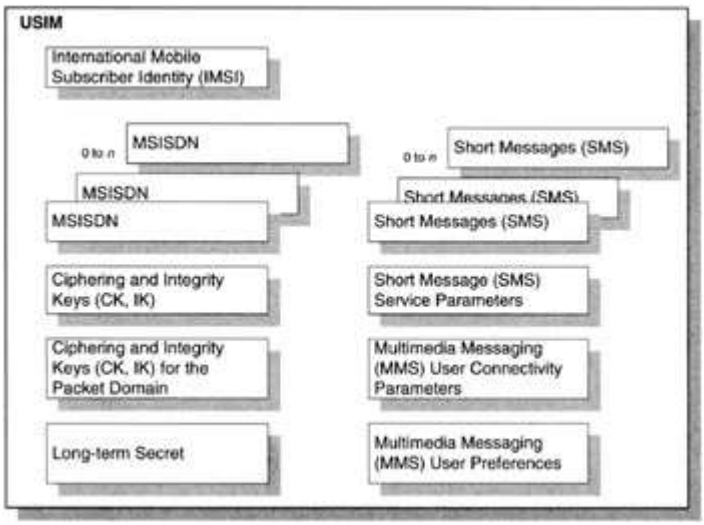

Figure 3.9 Simplified representation of the structure of the USIM application

- ✓ **Private User Identity:** ISIM stores the Private User Identity allocated to the user. There can only be one Private User Identity stored in ISIM.

- ✓ **Public User Identity:** ISIM stores one or more SIP URIs of Public User Identities allocated to the user.

- ✓ **Home Network Domain URI:** ISIM stores the SIP URI that contains the home network domain name. This is used to find the address of the home network during the registration procedure. There can only be one home network domain name URI stored in ISIM.

- ✓ **Long-term secret:** ISIM stores a long-term secret that is used for authentication purposes and for calculating the integrity and cipher keys used between the terminal and the network. The IMS terminal uses the integrity key to integrity protect the SIP signaling that the IMS terminal sends to or receives from the P-CSCF. If the signaling is ciphered, the IMS terminal uses the cipher key to encrypt and decrypt the SIP signaling that the IMS terminal sends to or receives from the P-CSCF.

All the above mentioned fields are read-only, meaning that the user cannot modify the values of the parameters. From the description of the fields contained in ISIM

the reader probably realized that ISIM is important to authenticate users. Access to a 3GPP IMS network relies on the presence of either an ISIM or a USIM application in the UICC. ISIM is preferred because it is tailored to the IMS although access with USIM is also possible. This allows operation in an IMS network of users who have not upgraded their UICCs to IMS-specific ones that contain an ISIM application.

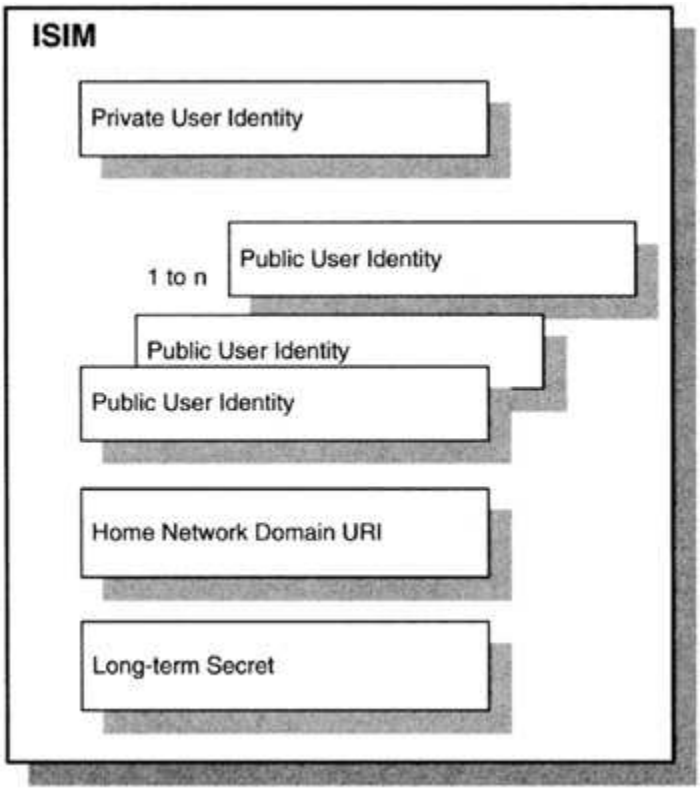

Figure 3.10 Structure of an ISIM application

Due to the lower degree of security contained in a SIM application, access to a 3GPP IMS network with a SIM application is not allowed. Non-3GPP IMS networks that do not support UICC in the IMS terminals (e.g., 3GPP2) store the parameters contained in the ISIM as part of the terminal's configuration or in the terminal's built-in memory.

3GPP2 IMS networks also allow to store the above mentioned parameters in an R-UIM (Removable User Identity Module). The R-UIM is a smart card secure storage, equivalent to a 3GPP UICC with an ISIM application.

The multiple access requirements introduce other means of access than GPRS. The IMS is just an IP network and, like any other IP network, it is lower layer and access-independent. Any access network can in principle provide access to the IMS. For instance, the IMS can be accessed using a WLAN (Wireless Local Access Networks), an ADSL (Asymmetric Digital Subscriber Line), an HFC (Hybrid Fiber Coax), or a Cable Modem.

Still, 3GPP, as a project committed to develop solutions for the GSM evolution, has focused on GPRS access (both in GSM and UMTS) for the first release of the IMS (i.e., Release 5). Future releases will study other accesses, such as WLAN.IMS terminals supporting audio capabilities are required to support the circuit-switched domain due to the inability of the IMS (at least in the first phases) to provide support for emergency calls. So, emergency calls are placed over the circuit-switched domain.

Chapter-4
SESSION INITIATION PROTOCOL

4.1 INTRODUCTION TO SIP AND THE INTERNET

The Session Initiation Protocol (SIP) is a new signaling protocol developed to set up, modify, and tear down multimedia sessions over the Internet[3]. This chapter covers some background for the understanding of the protocol. SIP was developed by the Internet Engineering Task Force (IETF) as part of the Internet Multimedia Conferencing Architecture, and was designed to dovetail with other Internet protocols such as TCP, UDP, IP, DNS, and others.

IMS relies on the session initiation protocol (SIP) for the development of applications and services. SIP is a signaling protocol specifically designed for multimedia. It offers advantages over signaling system 7 (SS7), which is used throughout the public switched telephone network (PSTN) and was designed specifically for voice services. Unlike SS7, SIP was designed to support voice, data, and multimedia services. SIP is focused on session control—establishing, changing and terminating sessions—and it supports dynamic modification of multimedia streams for any given session.

4.2 OVERVIEW OF SIP FUNCTIONALITY

SIP is an application-layer control protocol that 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. Media can be added to (and removed from) an existing session. 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.

SIP supports five facets of establishing and terminating multimedia communications:

1. *User location:* Determination of the end system to be used for communication.
2. *User availability:* Determination of the willingness of the called party to engage in communications.

3. *User capabilities:* Determination of the media and media parameter to be used.
4. *Session setup:* "ringing", establishment of session parameters at both called and calling party.
5. *Session management:* Including transfer and termination of sessions, modifying session parameters, and invoking services.

SIP is not a vertically integrated communications system. SIP is rather a component that can be used with other IETF protocols to build a complete multimedia architecture. Typically, these architectures will include protocols such as the Real-time Transport Protocol (RTP)[4] for transporting real-time data and providing QoS feedback, the Real-Time streaming protocol (RTSP) for controlling delivery of streaming media, the Media Gateway Control Protocol (MEGACO) for controlling gateways to the Public Switched Telephone Network (PSTN), and the Session Description Protocol (SDP) for describing multimedia sessions. Therefore, SIP should be used in conjunction with other protocols in order to provide complete services to the users. However, the basic functionality and operation of SIP does not depend on any of these protocols.

SIP does not provide services. Rather, SIP provides primitives that can be used to implement different services. For example, SIP can locate a user and deliver an opaque object to his current location. If this primitive is used to deliver a session description written in SDP, for instance, the endpoints can agree on the parameters of a session. If the same primitive is used to deliver a photo of the caller as well as the session description, a "caller ID" service can be easily implemented. As this example shows, a single primitive is typically used to provide several different services.

SIP can be used to initiate a session that uses some other conference control protocol. Since SIP messages and the sessions they establish can pass through entirely different networks, SIP cannot, and does not, provide any kind of network resource reservation capabilities. The nature of the services provided make security particularly important. To that end, 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. SIP works with both IPv4 and IPv6.

4.3 SIP ELEMENTS

SIP is an application-layer control protocol that handles the setup, modification, and tear-down of multimedia sessions. SIP is used in combination with other protocols to describe the session characteristics to potential session participants. SIP is based on a request and response transaction model similar to HTTP. Each transaction consists of a request that invokes a particular method or a function on the server and at least one response.

SIP is generally considered to be a Agent–server protocol. At a high level there are two types of SIP elements:

1. User Agents
2. Servers.

User agents: User Agents are endpoints in a SIP network: they originate and terminate calls. Examples of User Agents (UA)[5] include: SIP phones (hard sets), laptops or PDA with a SIP client (e.g., soft phone), Media gateway (e.g. T1/E1 gateway), access gateway (e.g., FAX gateway), conferencing systems, etc. All these devices also initiate and terminate the media session (voice, video, FAX, etc.). A UA is itself comprised of two entities (software):

• UAC (initiates call by sending INVITE with E.164 or URI dialing)

• UAS (receives call requests).

Server: A server generally responds to a request sent by a agent. A server can be a software application, such as Live Communications Server 2003, or a hardware device. There are several types of servers in a SIP network including

- Proxy server
- Redirect server
- SIP registrar.

4.3.1 Different roles of a SIP server

SIP servers have different roles, such as:

4.3.1.1 Proxy server: A Proxy server performs signaling and relay. In other words, it determines where to send signaling messages and forward requests on behalf of the UA. To do so, it consults databases (DNS, location servers, etc.)[6]. It is important to

remember that Proxy servers have no media capabilities; they are in the control path only. Proxy servers must pass unrecognized SIP messages through unchanged. Thus new features do not require changes to proxy servers used in an infrastructure. This principle enables new features to be deployed in a network by only upgrading the end devices. The routing function can be configured (programmed) according to user preferences, type of call (e.g., 911), least-GW-cost, or other criteria. Note that the proxy server is not the only "place" where service can be programmed. In fact, service programmability can reside in end-devices as well, such as for visual caller ID, distinctive ringing or possible Call Forwarding. Proxy servers can try several destinations sequentially or in parallel, this capability called forking enables multiple devices to be associated with the same address.

There are three types of Proxy servers according to the type of state information they keep:

1) A stateless proxy keeps no state

2) a transaction stateful proxy only keeps state on pending transactions.

3) a call stateful proxy keeps state for the entire duration of a SIP session.

Most implementations are stateful proxy-based as this is useful for implementing such services as "forward on no reply" and also to implement forking. Stateless proxies are easier to scale (especially under heavy load scenarios) and can act as an application-layer load distributor (used in the core of a network). Redundancy designs are easier to achieve with stateless proxies.

4.3.1.2 Redirect server: A SIP redirect server accepts a SIP request and conveys to the originating client the way to route the call. Redirect servers are servers that redirect SIP requests to another device. A redirect server responds to the request with the address to which the request should be redirected to (e.g., a request for nic@mitel.com can be redirected to nic@home.com). SIP does not specify any implementation models –for example, all above servers can reside on the same hardware platform. The underlying OS can be Windows, Solaris, Linux or any embedded real time OS. For example, VOCAL is an open-source VoIP [7] software from Vovida.org. VOCAL software suite is a robust implementation of the SIP protocol and its various entities and is used widely. It is important to note that the above servers (proxy, redirect and registrar) are all optional SIP components. In fact, a UA may issue an INVITE directly to a targeted endpoint and many

telephony features may be implemented directly on the UA. The SIP model is based on intelligent endpoints that can act without other intelligence from the network infrastructure (refer to section below on peer-to-peer vs. centralized model).

4.3.1.3 Registrar server: A SIP registrar server accepts registration requests and maps agent's address to a user's sign-in name, or SIP URI[8]. Typically, a registrar is combined with a proxy or redirect server. A SIP registrar accepts registration requests from users (e.g., I am now at 192.168.0.10) and maintains user location information in a database. Mobility is thus achieved by the use of a REGISTER message (from UA) and by keeping a location database updated.

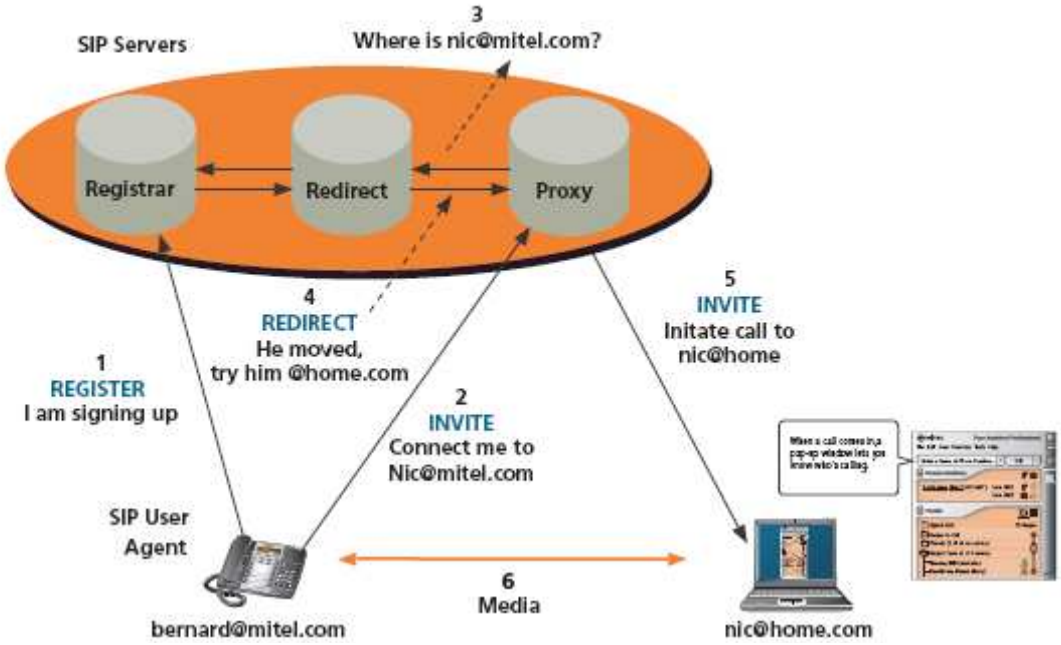

Figure 4.1 Example of user mobility using register and redirect messages

4.4 SIP MESSAGE STRUCTURE

SIP is a text-based protocol that is similar to HTTP, which makes it easy read and understand. A SIP message is either a request from a client to a server or a response from a server to a client. Both the request and the response contain a start-line followed by one or more headers and a message body. For example:

message = start-line

**message header*

CRLF

[message-body]

The request line specifies the type of request being issued, while the response line indicates the success or failure of a request. If a request is not executed, the status line indicates the type of failure or the reason for the failure.

- SIP messages are either a request or a response
- Structure of a SIP message includes:

  Start line
  Header(s)
  Body
- Sample structure

message = start-line

**message header*

CRLF

Figure 4.2 An overview of SIP message structure

4.4.1 SIP request

A SIP request consists of a method token, a request URI, and the SIP version. A method token is used to identify the request. The request URI is the address of the device where the request is being sent.

Methods for handling different kinds of requests

- INVITE
- ACK
- BYE
- CANCEL
- OPTIONS
- REGISTER

SIP method extensions include:

- SUBSCRIBE
- NOTIFY
- MESSAGE
- INFO
- SERVICE
- REFER
- NEGOTIATE

Figure 4.3 SIP request methods

The original SIP RFC 3261[9] defines six methods, which are used for different types of requests. The following table describes these methods.

Method Name	Description
INVITE	It initiates a session. This method includes information about the calling and called users and the type of media that is to be exchanged.
ACK	Sent by the client who sends the INVITE.ACK is sent to confirm that the session is established. Media can then be exchanged.
BYE	Terminate a session. This method can be sent be either user.
CANCEL	Terminates a pending request, such as an outstanding INVITE. After a session is established, a BYE method needs to be used to terminate the session.
OPTIONS	Queries the capabilities of the server or the other devices. It can be used to check media capabilities before issuing an INVITE
REGISTER	Used by a client to login and register its address with a SIP registrar server.

Table 4.1 Methods for handling different types of request

4.4.2 SIP method extensions

A number of extensions and enhancements have been made to the original SIP RFC 2543[10]. This includes the addition of the following new methods to SIP, which can be used for event notification, instant messaging and call control:

- **SUBSCRIBE**. The SUBSCRIBE method enables a user to subscribe to certain events. This means that the user should be informed when such events occur.
- **NOTIFY**. The NOTIFY method is used to inform the user that a subscribed event has occurred. Windows Messenger uses the SUBSCRIBE method to request contacts, groups, and allow and block lists from the server and to get the presence of contacts in a group. Live Communications Server 2003 uses the NOTIFY method to deliver the data obtained by the SUBSCRIBE method to the client.
- **MESSAGE**. SIP can also be used for Instant Messaging. A user sends an instant message to another user by sending a request that includes the MESSAGE method. This request carries the actual text in a body of a SIP packet.
- **INFO**. The INFO method is used for transferring information during a session, such as user activity. For example, Windows Messenger 5.0 uses the INFO method to inform the called user that Bob, the calling user, is typing on the keyboard. As a result, in the conversation UI, the called user sees a dialog, “bob is typing.”
- **SERVICE**. The SERVICE method can carry a Simple Object Access Protocol (SOAP) message as its payload. Windows Messenger 5.0 uses the SERIVCE method to add contacts and groups on the server. This method is also used to search for contacts in the SIP domain.
- **NEGOTIATE**. The NEGOTIATE method is used to negotiate various kinds of parameters, such as security mechanisms and algorithms. Live Communications Server 2003 uses the NEGOTIATE method to provide compression between clients and servers.
- **REFER**. A REFER request enables the sender of the request to instruct the receiver to contact a third party using the contact details provided in the request. Call Transfer is a commonly used application of the REFER method.

4.4.3 SIP Response

SIP response contains:

- Status code, three digit number indicating the outcome of the request
- Reason phrase, provides a textual description of the outcome

Different classes of a response:

- 1xx: provisional
- 2xx: success
- 6xx: global failure
- 3xx: redirection
- 4xx: client server
- 5xx: server error
- 6xx: global failure

A SIP respon that indicates the outcome of the request. The response also contains a reason phrase, which provides a textual description of the outcome of the request. The reason code is interpreted and acted upon by the client software. The reason phrase helps the user understand the response. Status codes defined in SIP have values between 100 and 699 and the first digit of the reason code indicates the response class. For example, all the status codes between 100 and 199 belong to one class.

The table 4.2 describes the different classes in SIP:

Class name	**Description**
1xx: Provisional	Request received, continuing to process the request. for example, 180 indicates that the phone of the called user is ringing.
2xx: Success	Action was successfully received, understood, and accepted. Only 200 OK and 202 ACCEPTED have been defined in this class
3xx: Redirection	Further action needs to be taken to complete the request. For example, a front-end server 302 to redirect the client to a home server.
4xx: Client Error	Request contains bad syntax or cannot be fulfilled at this server. For example a home server sends a response, 401 Unauthorized, if the client needs to provide credentials.

5xx: Server Error	Server failed to fulfill a valid request. For example a server sends a response, 504 timeout, if the MTLS has not been configured between the home servers.
6xx:Global failure	Request can not be fulfilled at any server. This is a new class defined for SIP, but is not currently used with live communication server 2003

Table 4.2 Different classes of SIP response[11]

4.4.4 SIP Headers

SIP includes a number of message headers in a SIP message. These headers contain information that enables the receiver to understand the message better or handle the message properly. Some headers make sense only in certain requests or responses. In some cases, the presence of a particular header depends on the context. The presence of a particular header in a response might be reasonable only if the response is issued to a specific request.

General headers:
- Used in both requests and responses.
- Contains basic information needed for the handling of requests and responses
- Examples: the To and From header fields

Request header:
- Apply only to SIP requests.
- Provide additional information to the server regarding the request itself or regarding the client.
- Examples: the Subject and Priority header fields

Response Header:
- Apply only to response (status) messages.
- Provide further information about the response that can not be included in the status line.
- Examples: unsupported and Retry after header fields

Figure4.5 Overview of SIP headers

4.4.4.1 General headers: Some headers can be used in both requests and responses. They are known as general headers[12]. Such headers contain basic information. For

example, the *To:* header field indicates the recipient of the request, *From:* indicates the originator of the request, and *Call-ID:* uniquely identifies a specific invitation to a session.

4.4.4.2 Request headers: Request headers apply only to SIP requests. They are used to provide additional information to the server about the request or the client. For example, *Subject:* can be used to provide a textual description of the topic of the session. *Priority:* is used to indicate the urgency of the request, such as emergency, urgent, normal, or non urgent.

4.4.4.3 Response headers: Response header fields apply only to response (status) messages. These header fields are used to provide further information about the response that cannot be included in the status line. For example, *Unsupported:* is used to identify those features that are not supported by the server. *Retry-After:* indicates when a called user will be available if the user is currently busy or unavailable.

4.5 A SIMPLE SIP EXAMPLE

Figure 4.6 shows the SIP message exchange between two SIP-enabled devices. The two devices could be SIP phones, hand-helds, palmtops, or cell phones. It is assumed that both devices are connected to an IP network such as the Internet and know each other's IP address[13].

The calling party, Tesla, begins the message exchange by sending a SIP INVITE message to the called party, Marconi. The INVITE contains the details of the type of session or call that is requested. It could be a simple voice (audio) session, a multimedia session such as a video conference, or it could be a gaming session.

The INVITE message contains the following fields:

INVITE sip:marconi@radio.org SIP/2.0

Via: SIP/2.0/UDP lab.high-voltage.org:5060

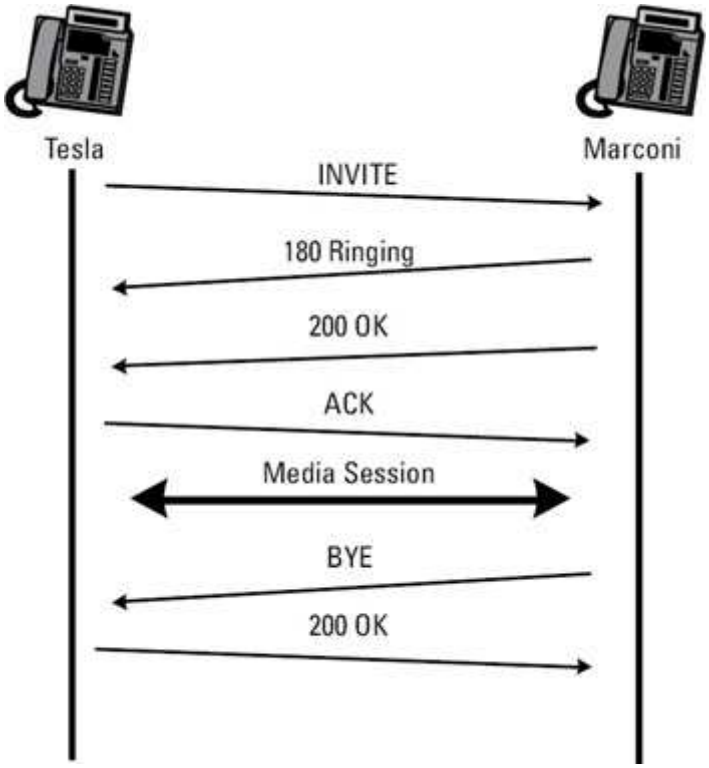

Figure 4.6 A simple SIP example.

To: G. Marconi <sip:Marconi@radio.org>

From: Nikola Tesla <sip:n.tesla@high-voltage.org>

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 INVITE

Subject: About That Power Outage...

Contact: sip:n.tesla@high-voltage.org

Content-Type: application/sdp

Content-Length: 158

v=0

o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org

s=Phone Call

c=IN IP4 100.101.102.103

t=0 0

m=audio 49170 RTP/AVP 0

a=rtpmap:0 PCMU/8000

The fields listed in the INVITE message are called headers. They have the form Header: Value CRLF. The first line of the request message, called the start line, lists the method, which is INVITE, the Request-URI (Uniform Resource Indicator), then the SIP version number (2.0), all separated by spaces. Each line of a SIP message is terminated by a CRLF[14]. The Request-URI is a special form of SIP URL and indicates the resource to which the request is being sent. SIP URLs.

The first header following the start line is a Via header. Each SIP device that originates or forwards a SIP message stamps its own address in a Via header, usually written as a host name that can be resolved into an IP address using a DNS query. The Via header contains the SIP Version number (2.0), a "/", then UDP for UDP transport, a space, then the hostname or address, a colon, then a port number, in this example the "well-known" SIP port number 5060.

The next headers are the To and From headers, which show the originator and destination of the SIP request. When a name label is used, as in this example, the SIP URL is enclosed in brackets and used for routing the request. The name can be displayed during alerting.

The Call-ID header has the same form as an e-mail address but is actually an identifier used to keep track of a particular SIP session. The originator of the request creates a locally unique string, then usually adds an "@" and its host name to make it globally unique. The combination of the local address (From header), remote address (To header), and Call-ID identifies the "call leg." The call leg is used by both parties to identify this call because they could have multiple calls set up between them. Subsequent requests for this call will refer to this call leg.

The next header shown is the CSeq, [15] or command sequence. It contains a number, followed by the method name, INVITE in this case. This number is incremented

for each new request sent. In this example, the command sequence number is initialized to 1, but it could start at another value.

The Via headers plus the To, From, Call-ID, and CSeq headers represent the minimum required header set in any SIP message. Other headers can be included as optional additional information, or information needed for a specific request type. A Contact header is included in this message, which contains the SIP URL of Tesla; this URL can be used to route messages directly to Tesla. The optional Subject header is present in this example. It is not used by the protocol, but could be displayed during alerting to aid the called party in deciding whether to accept the call. The same sort of useful prioritization and screening we all routinely do using the Subject and From headers in an e-mail message is also possible with a SIP INVITE request. Additional headers are present in this INVITE message, which contain the media information necessary to set up the call.

The Content-Type and Content-Length headers indicate that the message body is Session Description Protocol (SDP) [4] and contains 158 octets of data. A blank line separates message body from the header list, which ends with the Content-Length header. In this case, there are seven lines of SDP data describing the media attributes that the caller Tesla desires for the call. This media information is needed because SIP makes no assumptions about the type of media session to be established—the caller must specify exactly what type of session (audio, video, gaming) that he wishes to establish. The SDP field names are listed in Table 4.3. A quick review of the lines shows the basic information necessary to establish a session. This includes the:

- connection IP address (100.101.102.103);
- media format (audio);
- port number (49170);
- media transport protocol (RTP);
- media encoding (PCM μ Law);
- Sampling rate (8000 Hz).

SDP parameter	Parameter Name
v=0	Version number
o=Tesla 2890844526 2890844526 IN IP4 lab.high-voltage.org	Origin containing name
s=Phone Call	Subject
c=IN IP4 100.101.102.103	Connection
t=0 0	Time
m=audio 49170 RTP/AVP 0	Media
a=rtpmap:0 PCMU/8000	Attributes

Table 4.3 SDP data[4]

INVITE is an example of a SIP request message. There are five other methods or types of SIP requests currently defined in the SIP specification.

The next message in Figure 4.6 is a 180 Ringing message sent in response to the INVITE. This message indicates that the called party Marconi has received the INVITE and that alerting is taking place. The alerting could be ringing a phone, flashing a message on a screen, or any other method of attracting the attention of the called party, Marconi.

The 180 Ringing is an example of a SIP response message. Responses are numerical and are classified by the first digit of the number. A 180 response is an "informational class" response, identified by the first digit being a 1. Informational responses are used to convey non-critical information about the progress of the call. SIP response codes were based on HTTP version 1.1 response codes with some extensions and additions. Anyone who has ever browsed the World Wide Web has likely received a "404 Not Found" response from a web server when a requested page was not found. 404 Not Found is also a valid SIP "client error class" response in a call to an unknown user.

Response code number in SIP alone determines the way the response is interpreted by the server or the user. The reason phrase, Ringing in this case, is suggested in the standard, but any text can be used to convey more information. For instance, 180 Hold your horses, I m trying to wake him up! is a perfectly valid SIP response.

The 180 Ringing response has the following structure:

SIP/2.0 180 Ringing

Via: SIP/2.0/UDP lab.high-voltage.org:5060

To: G. Marconi <sip:marconi@radio.org>

From: Nikola Tesla <sip:n.tesla@high-voltage.org>

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 INVITE

Content-Length: 0

The message was created by copying many of the headers from the INVITE message, including the Via, To, From, Call-ID, and CSeq, then adding a response start line containing the SIP version number, the response code, and the reason phrase. This approach simplifies the message processing for responses.

Note that the To and From headers are not reversed in the response message as one might expect them to be. Even though this message is sent to Marconi from Tesla, the headers read the opposite. This is because the To and From headers in SIP are defined to indicate the direction of the request, not the direction of the message. Since Tesla initiated this request, all messages will read To: Marconi From: Tesla.

When the called party decides to accept the call (i.e., the phone is answered), a 200 OK response is sent. This response also indicates that the type of media session proposed by the caller is acceptable. The 200 OK is an example of a "success class" response. The 200 OK message body contains Marconi's media information:

SIP/2.0 200 OK

Via: SIP/2.0/UDP lab.high-voltage.org:5060

To: G. Marconi <sip:marconi@radio.org>

From: Nikola Tesla <sip:n.tesla@high-voltage.org>

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 INVITE

Contact: sip:marconi@radio.org

Content-Type: application/sdp

Content-Length: 155

v=0

o=Marconi 2890844526 2890844526 IN IP4 tower.radio.org

s=Phone Call

c=IN IP4 200.201.202.203

t=0 0

m=audio 60000 RTP/AVP 0

a=rtpmap:0 PCMU/8000

This response is constructed the same way as the 180 Ringing response. The media capabilities, however, must be communicated in a SDP message body added to the response. From the same SDP fields as Table 4.3, the SDP contains:

- end-point IP address (200.201.202.203);
- media format (audio);
- port number (60000);
- media transport protocol (RTP);
- media encoding (PCM μ Law);
- sampling rate (8000 Hz).

The final step is to confirm the media session with an "acknowledgment" request. The confirmation means that Tesla can support the media session proposed by Marconi.

This exchange of media information allows the media session to be established using another protocol, RTP in this example.

ACK sip:marconi@radio.org SIP/2.0

Via: SIP/2.0/UDP lab.high-voltage.org:5060

To: G. Marconi <sip:marconi@radio.org>

From: Nikola Tesla <sip:n.tesla@high-voltage.org>

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 ACK

Content-Length: 0

The command sequence, CSeq, has the same number as the INVITE, but the method is set to ACK. At this point, the media session begins using the media information carried in the SIP messages. The media session takes place using another protocol, typically RTP.

This message exchange shows that SIP is an end-to-end signaling protocol. A SIP network or SIP server is not required for the protocol to be used. Two end-points running a SIP protocol stack and knowing each other's IP addresses can use SIP to set up a media session between them.

Although less obvious, this example also shows the client-server nature of the SIP protocol. When Tesla originates the INVITE request, he is acting as a SIP client. When Marconi responds to the request, he is acting as a SIP server. After the media session is established, Marconi originates the BYE request and acts as the SIP client, while Tesla acts as the SIP server when he responds. This is why a SIP-enabled device must contain both SIP server and SIP client software—during a typical session, both are needed. This is quite different from other client-server Internet protocols such as HTTP or FTP. The web browser is always an HTTP client, and the web server is always an HTTP server, and similarly for FTP. In SIP, an endpoint will switch back and forth during a session between being a client and a server.

In Figure 4.6, a BYE request is sent by Marconi to terminate the media session:

BYE sip:n.tesla@high-voltage.org SIP/2.0

Via: SIP/2.0/UDP tower.radio.org:5060

To: Nikola Tesla <sip:n.tesla@high-voltage.org>

From: G. Marconi <sip:marconi@radio.org>

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 BYE

Content-Length: 0

The Via header in this example is populated with Marconi's host address. The To and From headers reflect that this request is originated by Marconi, as they are reversed from the messages in the previous transaction. Tesla, however, is able to identify the call leg and tear down the correct media session.

The confirmation response to the BYE is a 200 OK:

SIP/2.0 200 OK

Via: SIP/2.0/UDP tower.radio.org:5060

To: Nikola Tesla <sip:n.tesla@high-voltage.org>

From: G. Marconi <sip:marconi@radio.org>

Call-ID: 123456789@lab.high-voltage.org

CSeq: 1 BYE

Content-Length: 0

The response echoes the CSeq of the original request: 1 BYE.

4.6 REGISTERING TO IMS

Figure 4.7 below depicts in deep details how the subscriber registers to the IMS. Basic SIP knowledge is required.

1. GPRS Attach / PDP Context Establishment and P-CSCF Discovery (UE to GPRS). This signalling flow is shown to indicate prerequisites for the registration signaling.

2. REGISTER request (UE to P-CSCF). The purpose of this request is to register the user's SIP URI with a S-CSCF in the home network. This request is routed to the P-CSCF because it is the only SIP server known to the UE. In the following SIP request, the Contact field contains the user's host address. The P-CSCF will perform two actions, binding and forwarding.

3. DNS: DNS-Q based on the user's URI, the P-CSCF determines that UE is registering from a visiting domain and performs the DNS queries to locate the I-CSCF in the home network. The look up in the DNS is based on the address specified in the Request URI. The P-CSCF sends the REGISTER request - after local processing - to the address indicated in the Request- URI. When forwarding the REGISTER request the P-CSCF needs to specify the protocol, port number and IP address of the I-CSCF server in the home network to which to send the REGISTER request. The P-CSCF tries to find this information by querying the DNS.

4. REGISTER request (P-CSCF to I-CSCF). The P-CSCF needs to be in the path for all mobile originated and mobile terminated requests for this user. The P-CSCF binds the public user identity under registration to the Contact header supplied by the user. Cx: User registration status query procedure The I-CSCF makes a request for information related to the subscriber registration status by sending the private user identity, public user identity and visited domain name to the HSS. The HSS returns the S-CSCF required capabilities and the I-CSCF uses this information to select a suitable S-CSCF.

5. REGISTER request (I-CSCF to S-CSCF).This signaling flow forwards the REGISTER request from the I-CSCF to the S-CSCF selected.

6. Cx: Authentication procedure As the REGISTER request arrived without integrity protection to the P-CSCF, the S-CSCF shall challenge it. For this, the S-CSCF requires at least one authentication vector to be used in the challenge to the user. If a valid AV is not available, then the S-CSCF requests at least one AV from the HSS. The S-CSCF indicates to the HSS that it has been assigned to serve this user.

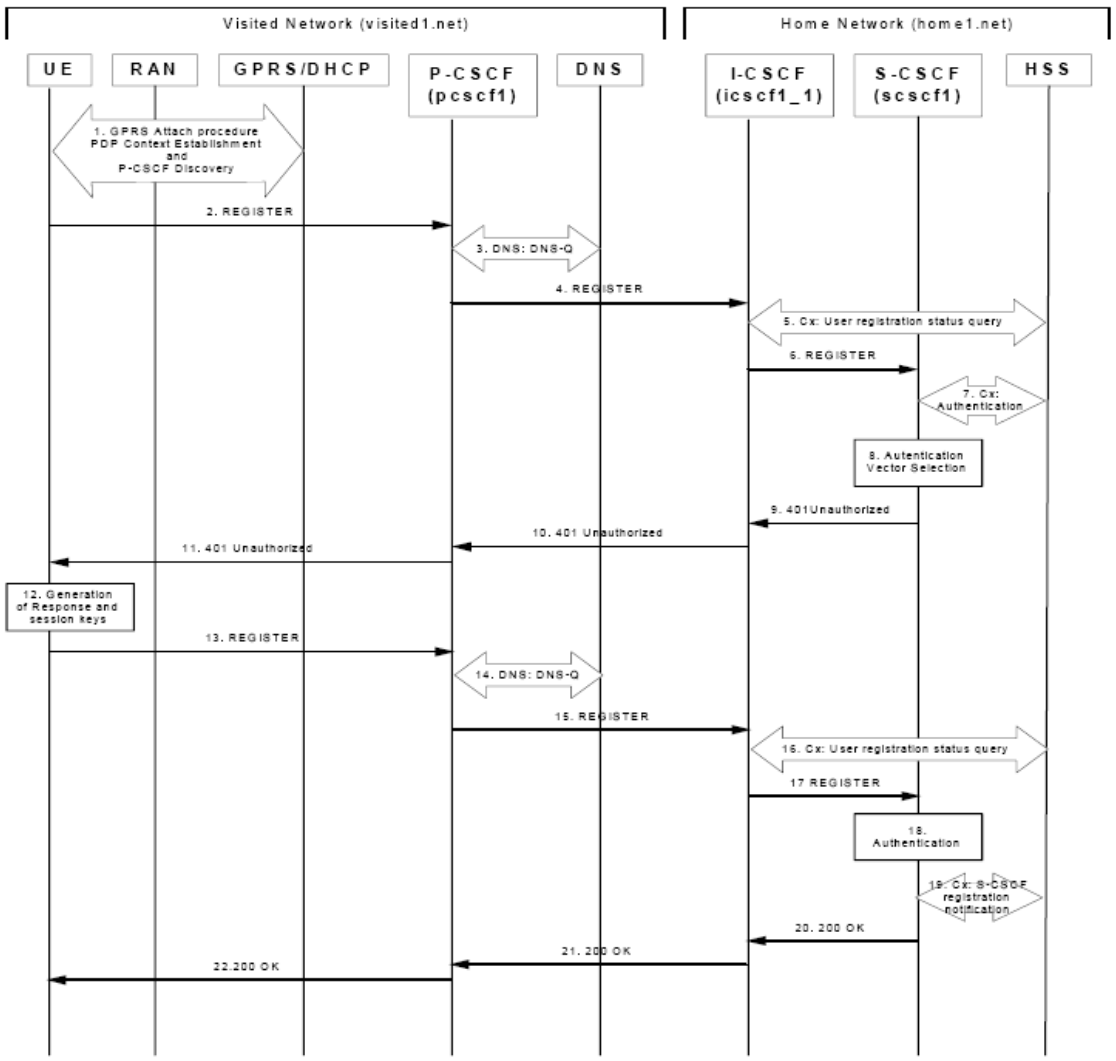

Figure 4.7 Registration to IMS using SIP

7. Authentication vector selection The S-CSCF selects an authentication vector for use in the authentication challenge.

8. 401 Unauthorized response (S-CSCF to I-CSCF) The authentication challenge is sent in the 401 Unauthorized response towards the UE.

9. 401 Unauthorized response (I-CSCF to P-CSCF) The authentication challenge is sent in the 401 Unauthorized response towards the UE.

10. 401 Unauthorized response (P-CSCF to UE).The P-CSCF removes any keys received in the 401 Unauthorized response and forwards the rest of the response to the UE.

11. Generation of response and session keys at UE Upon receiving the unauthorized response, UE calculates the response, and sends the response in the REGISTER SIP request.

12. REGISTER request (UE to P-CSCF)

13. DNS: DNS-Q based on the user's URI, the P-CSCF determines that UE is registering from a visiting domain and performs the DNS queries to locate the I-CSCF in the home network. The look up in the DNS is based on the domain name specified in the Request URI. The P-CSCF sends the REGISTER request - after local processing - to the address indicated in the Request-URI. When forwarding the REGISTER request the P-CSCF needs to specify the protocol, port number and IP address of the I-CSCF server in the home network to which to send the REGISTER request. The P-CSCF tries to find this information by querying the DNS.

14. REGISTER request (P-CSCF to I-CSCF)

15. Cx: User registration status query procedure The I-CSCF requests information related to the subscriber registration status by sending the private user identity, public user identity and visited domain name to the HSS. The HSS returns the S-CSCF name which was previously selected in step 5 (Cx: User registration status query procedure).

16. REGISTER request (I-CSCF to S-CSCF). This signaling flow forwards the REGISTER request from the I-CSCF to the S-CSCF.

17. Authentication: upon receiving an integrity protected REGISTER request carrying the authentication response, S-CSCF checks the response. If the check is successful then the user has been authenticated and the public user identity is registered in the S-CSCF.

18. Cx: S-CSCF registration notification procedure. On registering a user the S-CSCF informs the HSS that the user has been registered at this instance. Upon being requested by the S-CSCF, the HSS will also include the user profile in the response sent to the S-CSCF.

19. 200 OK response (S-CSCF to I-CSCF) The S-CSCF sends acknowledgement to the I-CSCF indicating that Registration was successful.

20. 200 OK response (I-CSCF to P-CSCF) .The I-CSCF forwards acknowledgement from the S-CSCF to the P-CSCF indicating that Registration was successful.

21. 200 OK response (P-CSCF to UE). The P-CSCF forwards acknowledgement to the UE indicating that Registration was successful. After the successful registration user must subscribe to the registration state event package in order to receive registration specific event notifications.

4.7 TOOLS TO READ SIP MESSAGES

The SIP Logger and SIP Parser tools can be used to read a SIP message.

4.7.1 SIP Logger:

SIP Logger is an executable file with a readme file, LoggerReadme.htm that contains information about the installation and use of SIP Logger. SIP Logger allows the Live Communications Server administrator to log SIP traffic on the server before it is encrypted. SIP Logger helps the administrator debug communication issues between clients and servers and between multiple servers. Messages are sent to a text file that is selected during SIP Logger startup[16]. The default maximum size of the file is 100 MB. Note that SIP Logger is not intended to replace the Live Communications Server IM Archiving Server.

- SIP Logger
    1. It is an .exe file that allows the administrator to log SIP traffic on the server before it is encrypted.
    2. Provide an aid to the administrator for debugging communication issues between clients and servers.
    3. Messages are dumped to a text file selected at startup of the SIP Logger
- SIP Parser
    1. It is a DLL file that works with Network Monitor to enable parsing of SIP signaling information exchanged between clients and servers.
    2. Captures SIP messages sent over TCP or UDP, TLS is not supported

4. SIP Parser is a dynamic-link library (DLL) that works with Network Monitor to enable parsing of the SIP signaling information exchanged between clients and servers. SIP Parser captures only SIP messages sent over TCP or UDP. TLS is not supported

because of encryption. SIP Parser requires the installation of Network Monitor on the computer where SIP messages need to be parsed. A readme file, SipParserReadme.html, contains information about the installation and use of SIP Parser.

Chapter-5
IMPLEMENTATION OF INSTANT MESSAGING & PRESENCE

5.1 INTRODUCTION

This chapter covers the new concepts of presence and instant communications, and shows how they can be implemented using SIP in IP multimedia subsystems. Polite calling, automatic call-back, avoiding unsuccessful calls, and legitimate tracking of specialized workforce members are some of the new communication services enabled by presence and using SIP.

5.2 THE EMERGENCE OF INSTANT MESSAGING

It is not practical to write the very short history of the emergence of instant messaging (IM) on the Internet, since any data would be obsolete by now. IM is, at present, the technology, competitive, and regulatory battleground for the largest and smallest software companies and service providers alike. In this chapter the technology concepts on which instant messaging [17] and its successor, instant communications (IC)[18], are based, and describe how IC can be implemented with SIP.

The first widespread use of IM was AOL's Instant Messenger, which proved to be so popular that many non-AOL customers signed up for a free IM account. The companion "Buddy List," which allows a user to be notified when a specified set of users is active, also represents a primitive presence client. However, the first IM products used proprietary protocols and centralized server architecture. Efforts by various IM developers to internetwork have not been successful. As a result, there has been a strong push in the industry to develop an open standard, interoperable, and scalable protocol for IM. This has lead to the formation of the IETF Instant Messaging and Presence Protocol Working Group (IMPP WG). To date, this group has produced two key documents on requirements and a model for presence and instant messaging. It soon became apparent, however, that:

1. The newly discovered presence service may be used for all other communication services, beyond short text messaging.
2. IM by itself can be implemented using various protocols. Three contending protocols emerged in the IMPP WG:
    a) SIP for general communications applications, or SIMPLE (SIP for Instant Messaging and Presence), as the extensions to SIP are known.
    b) IMXP [19] to keep IM the simplest it can be and to build on email,
    c) PRIM [20] also to keep IM lightweight, but using TCP transport.

The commonalities and differences were clearly articulated and it was felt the different approaches may meet different needs and should have only a common model and data exchange format for interoperability between the various protocols. Another key document, the Common Profile for Instant Messaging (CPIM)[19], was the result of this agreement in the IMPP WG.

In conclusion, the internal protocols and data formats of various IM systems are a local design decision, but interoperability between IM systems should be possible via CPIM.

5.3THE IETF MODEL FOR PRESENCE AND INSTANT MESSAGING

Presence and instant messaging are made possible by the packet nature of the Internet and may merit dedicated tutorials on their own. An attempt is made here only to give some basic notions that help in understanding the new service and it's potential. The model for a presence service is shown in Figure 5.1a.

The model for instance instant messaging is similar and is shown in Figure 5.1b. Both services have other similarities such as the notions of *principals* that can be either people or software that appear to the service as a single entity. Principals interact with the system via *user agents.* A user agent is the coupling between the principal and some core entity in the system.

The document defines a standard data format for presence, which is composed of so-called presence tuples. Each presence tuple consists of the following fields:

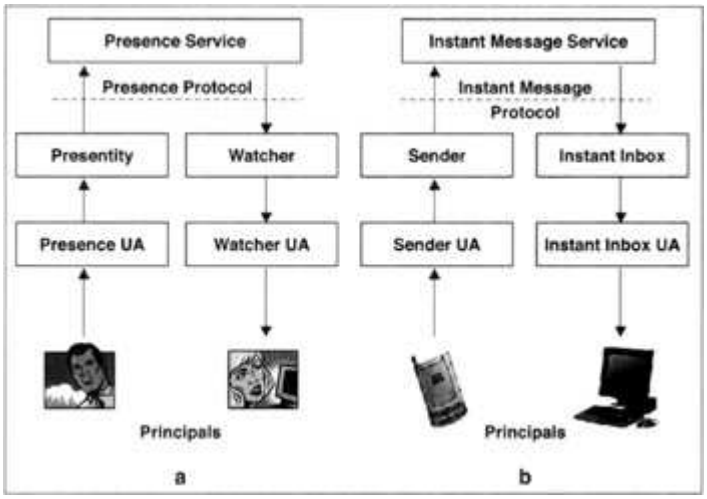

Figure 5.1 Models for presence and instant messaging

- **Status.** Online, offline, busy, away, do not disturb
- **Communication address.** Includes the:
    - **Contact means.** Such as messaging (short, email), pager, PSTN, etc.
    - **Contact address.** The service-specific URL
- **Other Markup.** Not yet specified.

The document defines a *presentity* as the software that provides presence information to the presence service. While the presence service handles distribution of the information, it is the presentity that generates a message called a *notification* about the presence information of the principal. Contained in the notification is the *status* of the principal, defined in the document as *open* or *closed*, or other mutually exclusive values. The nature of these status values depends on the nature of the service. Requests to the presentity are sent by a *watcher.* These terms are defined as follows:

- **Watcher.** Requests presence information about one or more presentities or about other watchers from the presence service. Special types of watchers are:
    - **Subscriber:** Asks the presence service to be immediately notified of any changes to one or more presentities.
    - **Fetcher:** Makes a request for presence information, but has not requested a subscription to the presence service.
    - **Poller:** Is a fetcher that makes regular requests to update presence information?

- **Notification.** Is a message sent from the presence service to a subscriber when there is a change of presence information by some presentity of interest to the watcher?
- **Status.** Is a distinguished part of the presence information about presentity. Status can have at least two mutually exclusive values: *open* or *closed.* Open or closed has meaning for instant messaging and there may be equivalent notions for other means of communications such as free or busy in circuit switched telephony. Other means of communications also can have different status values, in addition to open or closed.
- **Presence service.** Accepts, stores, and distributes presence information.
- **Instant message service.** Accepts and delivers instant messages.

Both the presence and the instant message services may have complex internal structures with specific servers and/or proxies with quite complex security implementations. In keeping with the end-to-end control principle of the Internet, these services also can be implemented in the endpoints, without dependence on intermediate elements in the network, as is the case with SIP.

5.4 SECURITY FOR PRESENCE AND IM

Security considerations for presence and instant messaging deal with:

- **Spam:** Unwanted instant messages. Delivery rules are intended to deal with Spam.
- **Spoofing:** The imitation of a principal by another principal. Authentication rules are intended to deal with spoofing.
- **Stalking:** Using presence information about the whereabouts of a principal for illegal or malicious purposes. Access rules, visibility rules, and rules for the distribution of watcher information are intended to deal with stalking.

5.5 THE COMMON PROFILE FOR INSTANT MESSAGING

As mentioned, IM systems can use different protocols and different data formats, but should meet the definition of the Common Profile for Internet Messaging (CPIM) for

interoperability. CPIM interoperability is expressed in terms of an abstract presence service and an abstract instant message service.

The documents define a new URL scheme—"im"—which represents the resource of the specified user's instant message inbox. The addresses use the familiar e-mail form of user@host or user@domain. The URL of an IM recipient could be, for example: im:student@college.edu.

Note that this URL does not define the transport protocol. As a result, the IP address lookup for the URL depends on the particular transport protocol used by the local IM system. If SIP is used for transport, the DNS resource record (RR) is found by executing a service lookup (SRV) for the address of the SIP proxy for the *college.edu* domain to determine the next hop for the message.

The abstract models for presence and instant message services in CPIM are of a very simple nature and are represented in Figure 5.2.

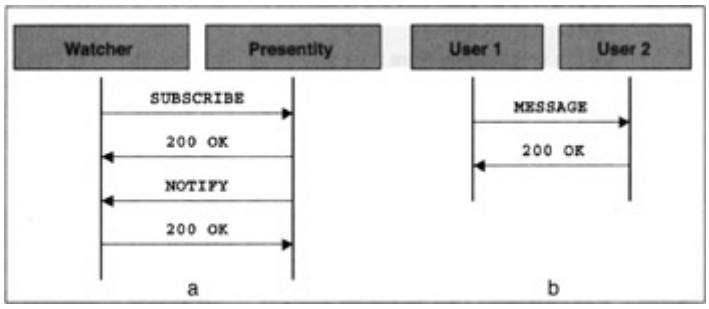

Figure 5.2 CPIM models for presence and instant message services.

Figure 5.2a shows the message exchange to subscribe and Figure 5.2b shows the basic message exchange for a subscribed user. There also is a corresponding *Unsubscribe* message, not shown in Figure 5.2.

Message flows as specified in CPIM, are reproduced here, since they illustrate the minimalist requirements for interoperability. HTML is used in the example, though it is not a requirement to be used between systems or inside any particular system. Gateways may be necessary if the interworking systems use different data representations, though the systems may internetwork directly, depending on their implementation.

5.6 PRESENCE SERVICE

A watcher friend1 subscribes to the presence service associated with presentity of friend2. The requested time in the example is 24 hours (86,400 seconds), but only 1 hour is returned in the response.

Subscribe

The subscribe is:

```
<subscribe watcher= 'pres:friendl@ispl.com'
                target='pres:friend2@isp2.com'
                 duration='64000' transID='l' />
```

Response

The response is:

```
<response status='success' transID='l' duration='3600' />
```

The successful subscription will enable the notify operation to communicate the presentity information to the watcher.

Notify

The notify is:

```
<notify watcher='pres:friendl@ispl.com'
            target='pres:friend2@isp2.com'
            transID='l' />
  <presence entityInfo='http://www.isp2.com/friend2' />
    <tuple destination='im:friend2@isp2.com'
                status='open' />
</notify>
```

A watcher can unsubscribe from the presence service (not shown in Figure 5.2):

Unsubscribe

The unsubscribe is:

```
<unsubscribe watcher='pres:friendl@ispl.com'
             target='pres:friend2@isp2.com'
             transID='l' />
```

and, if successful, will be informed:

```
<reponse status='success' transID='l' />
```

5.7 INSTANT MESSAGE SERVICE

The watcher can now send a message, knowing the open presence status of the inbox.

Message

The message is:

```
<message source='im:john.doe@ispl.com'
    destination='im:mary.king@isp2.com'
    transID='l' />
Content-Type: text/plain; charset="us-ascii"
Hello! How are you?
```

Response

The response is:

```
<response status='success' transID='l' />
```

In case of success, the response will be:

```
<response status= 'success' transID='l' />
```

Though of minimalist nature, CPIM interworking between instant message systems will still meet the requirements described in RFC 2778 for interoperability

5.8 WHY SIP FOR PRESENCE AND INSTANT MESSAGING?

At some time, there is an abundance of instant messaging products and services on the market, mostly implemented on centralized servers. None of these IM services interoperate, and none uses a standard, open protocol. The development of the requirements for a standard, open, interoperable protocol in the IETF [23] has resulted in multiple standards proposals in the IMPP WG, out of which only one was based on SIP. Why use SIP for presence and instant messaging? There are a number of reasons to choose SIP for commercial grade presence and instant communications:

- Presence is useful for any type of communications, not only short text messaging.
- SIP already solves the tasks required for presence and instant communications:
    - User agent registration
    - User agent authentication
- Rendezvous between parties via call routing. SIP call setup and presence are dual aspects of the rendezvous feature. In call setup, the message is routed from the caller to the called party at the request of the caller, while in presence the notification of status change is routed to the watcher, whenever it happens.
- Use of existing infrastructure. Clients and servers, software and databases, and, last but not least, the security mechanisms deployed for SIP.
- SIP has a decentralized, highly scalable architecture.

A number of IETF drafts, each of them dealing with a particular topic, have provided the complete information to build presence and IM using SIP[24]. Basic concepts of the work done are illustrated here.

SIP requires three methods, called SUBSCRIBE, NOTIFY, and MESSAGE to support presence and instant communications of any form such as text, voice, data, video, or for interactive games.

5.9 ARCHITECTURE OF IMPS

The architecture of the IMPS service is depicted in Figure 5.3. IMPS is based on a client server architecture, all traffic sent from a client passes through the server, peer-to-peer messaging is not supported.

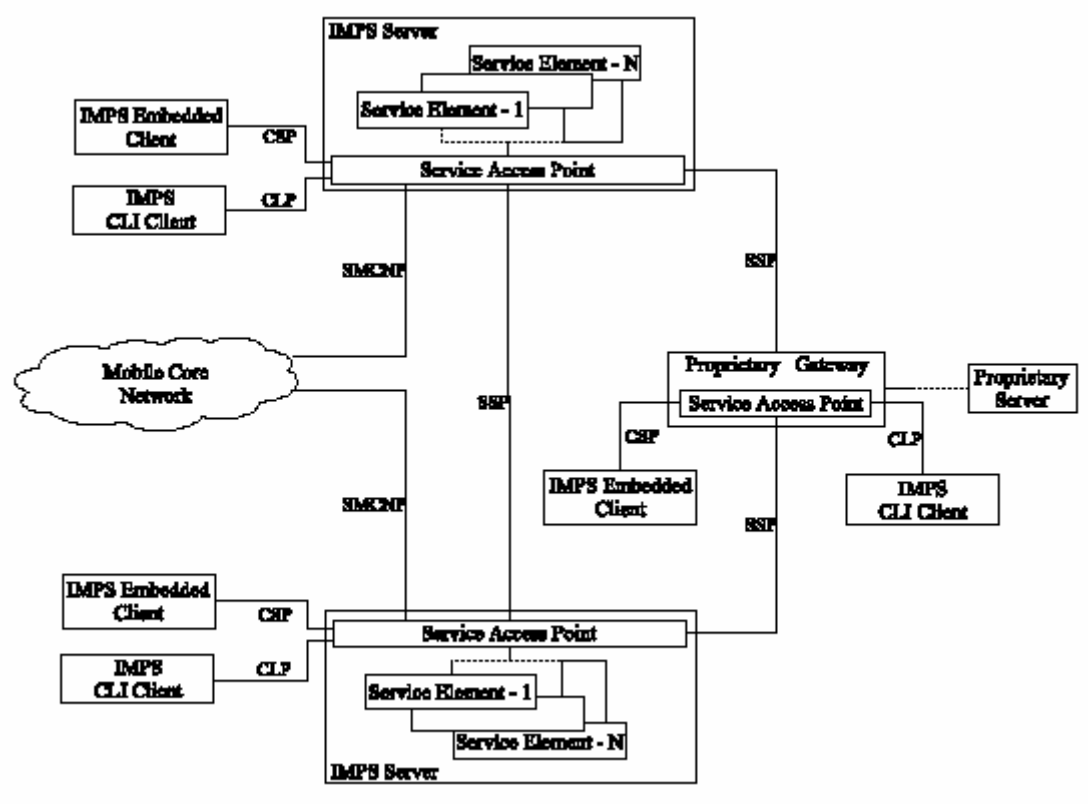

Figure 5.3: IMPS architecture [6]

5.9.1 IMPS Server

The IMPS server holds five important elements; the Service Access Point (SAP) and four service elements. The SAP serves as the interface through which the outside environment can communicate with the IMPS server. The interface provides IMPS clients, the mobile core network, proprietary gateways and other IMPS servers with access to the functionality of the SAP and the service elements. The functionality of the SAP includes:

- Authentication and authorization
- Service discovery and service agreement
- User profile management
- Service relay

The functionality specific to instant messaging can be divided into four logical groups. Each service element comprises the functionality of one such group. Table 5.1 lists the service elements and their main functionality.

Service element	Main functionality
Presence	Presence management Presence subscriptions Presence authorization Contact list management
Instant messaging	Instant message delivery Access control

Table 5.1 IMPS service elements

All IMPS servers are required to provide SAP functionality but service elements can be scattered among several servers; a server is not required to implement all of them. This Chapter facilitates the creation of distributed IMPS services, where servers relay requests to the server containing a particular service element using the Server-Server Protocol (SSP).

5.9.2 IMPS Clients

The IMPS system defines two types of clients, Embedded Clients and CLI (Command Line Interface) Clients. The Embedded Client can be embedded into several different environments, e.g. mobile terminals, fixed PC-clients and automated applications. The Client-Server Protocol (CSP) allows these clients to be fully interoperable despite their differences. CLI Clients use the text message based Command Line Protocol (CLP) to communicate with IMPS servers. CLP consists of commands typed by the user and sent as SMS messages to the IMPS server, which sends an SMS message in reply for the user to read and interpret. Consequently, CLI Clients need no software except for the ability to send and receive SMS messages. CLI Clients provide only a subset of the functionality provided by Embedded Clients.

5.10 ARCHITECTURE DETAILS OF SIMPLE

SIMPLE builds upon the SIP protocol, and much of the underlying technology used to locate resources, route messages, and establish sessions is shared between SIP and SIMPLE. SIP uses the notion of a proxy to locate and provide name resolution services. Figure 5.3 shows the architecture of SIMPLE. As mentioned previously, the SIMPLE specifications are not yet finalized and therefore changes to the architecture,

such as addition or removal of reference points, are possible. The reference points and their associated communication protocols are summarized in Table 5.1 and further elaborated on in the following subsections. As can be seen from the table, protocols providing direct interaction between the server elements are still to be defined. Some of these undefined reference points are not necessarily needed as server elements can communicate with each other utilizing reference points used by clients. In this case a server element acts as a client in order to use the services provided by another server. For example, a GLMS (Group and List Management Server) can subscribe to presence information provided by a presence server. Furthermore, several server elements can be co-located into one physical element, in which case an element has direct access to the information of the other co-located elements.

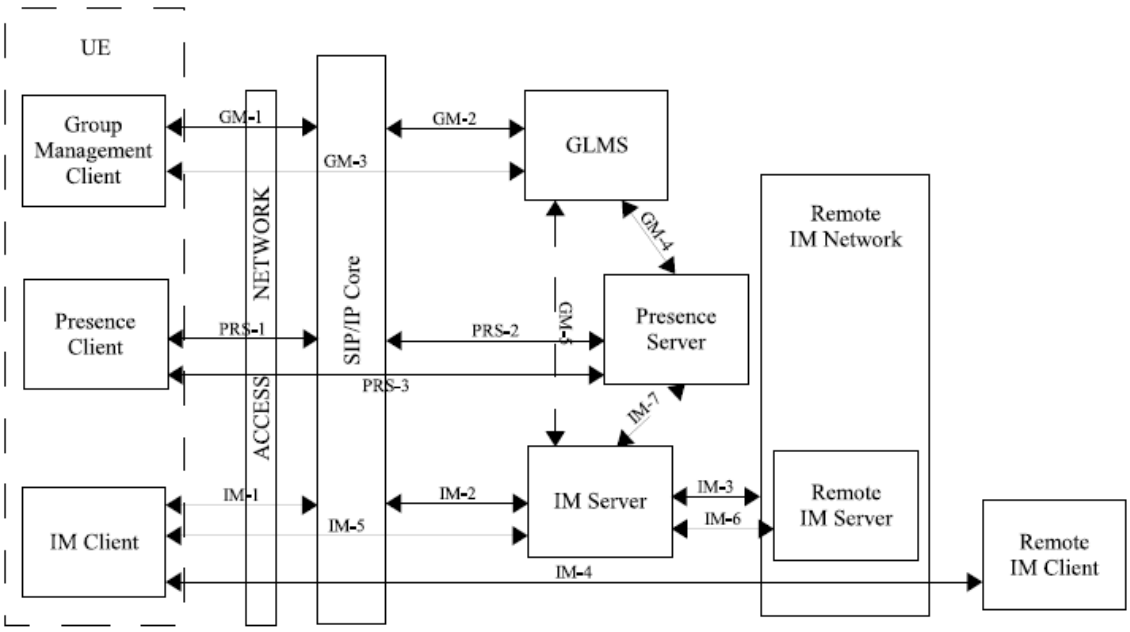

Figure 5.4 SIMPLE architecture

It should be noted that Figure 5.4is somewhat simplified as the server elements themselves can consist of several smaller elements that are not required to exist at the same location either (for an example, see Figure 5.7).

Reference point	Functions	Protocol

IM-1,IM-2,IM-3	Session signaling between IM elements using the SIP/IP core	SIP
IM-4	Peer to peer IM services	MSRP
IM-5,IM-6	Messaging through IM services	MSRP
IM-7	Communication between a presence server and IM server	Undefined
PRS-1,PRS-2	Communication between a presence client and a presence server	SIP
PRS-3	Presence information and authorization management	XCAP
GM-1,GM-2	Communication between a group management client and GLMS	SIP
GM-3	Management of groups and lists at the GLMS	XCAP
GM-4	Communication between a GLMS and a Presence server	Undefined
GM-5	Communication between a GLMS and an IM server	undefined

Table 5.2 SIMPLE reference points

However no means for communication between the sub-elements of the server elements have been defined. In practice, it follows that the sub-elements usually will be co-located in the same physical entity, similar to the servers of Figure 5.3.

5.10.1 IM Server

SIMPLE provides two modes for sending instant messages.

- pager mode
- session mode.

In pager mode instant messages are sent over SIP using the MESSAGE method extension as described in figure 5.4[14]. When using session mode messaging, a session is established using SIP, after which the Message Session Relay Protocol (MSRP) is used for exchanging instant messages within the session as shown in figure 5.5[13].

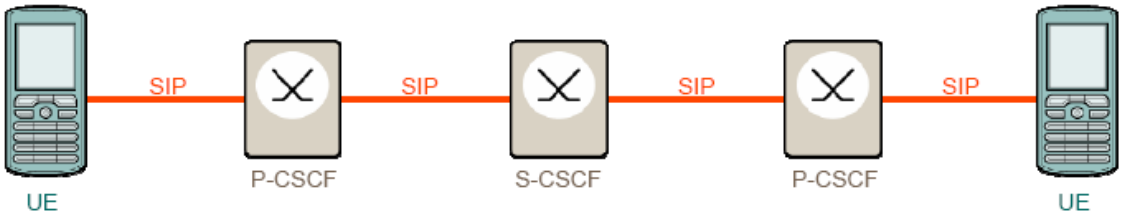

Figure 5.5 Pager mode messaging service

Both modes are able to function without any actual server functionality. In this case the IM Server element of Figure 5.4 simply consists of a regular SIP proxy. When using pager mode, IM Server elements only route the message to the recipient over IM-1, IM-2 and IM-3. For session mode messaging, the session is initiated using IM-1, IM-2 and IM-3, while the actual message session is established over IM-4 using MSRP. MSRP sessions can also be directed through MSRP relay elements using reference points IM-5 and IM-6. Currently this is the only case where the IM Server could include extra functionality. For example, a store-and-forward mechanism enabling offline messaging could be implemented

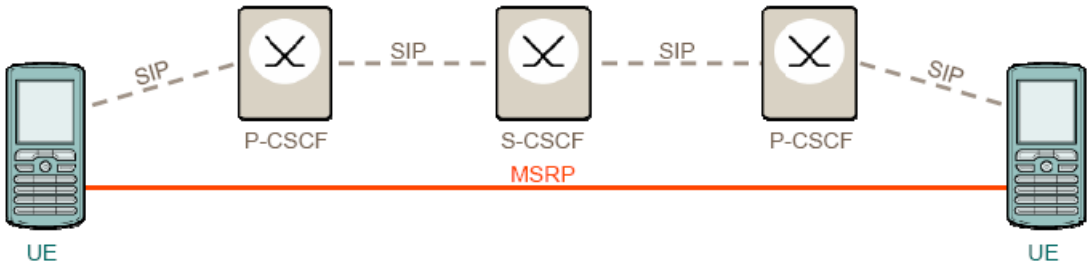

Figure 5.6 Session mode paging service

5.10.2 Presence Server

By definition a SIMPLE presence server is a physical entity either acting as a presence agent or forwarding incoming subscriptions to entities that may act as presence agents[58]. A presence agent is a SIP user agent, which manages presence subscriptions and sends notifications to watchers whenever the presence information of the subscribed presentity is updated. In order to manage presence subscriptions and notifications the presence agent needs access to both presence information and presence authorization rules, both defined as separate entities. As the methods to be used for communicating

between these entities are undefined, most implementations will co-locate all in one physical element, generally called the Presence Server, as shown in Figure 5.6

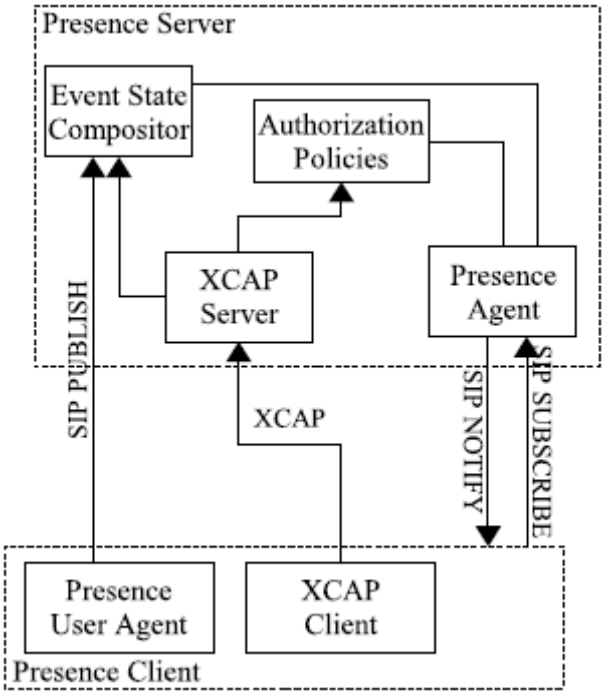

Figure 5.7 SIMPLE Presence Server

Figure 5.7 also illustrates another aspect typical to SIMPLE; on several occasions multiple methods for performing the same function exist. For example, presence information can be updated either using the SIP PUBLISH method over PRS-1 and PRS-2 or using the XML Configuration Access Protocol (XCAP)[59] over PRS-3.

5.10.3 GLMS

The Group and List Management Server (GLMS) is responsible for the management of contact lists, group lists and group access lists. The GLMS allows users to create groups and to define the users which are allowed access to the created groups. The XCAP protocol (over GM-3) is generally used to manage the content on the GLMS. The GLMS might act as a server for ongoing group messaging sessions as well, but group messaging sessions can also be hosted by other entities, such as dedicated application servers or the hosts that created the groups. Table 5.2 lists the main SIP methods and their use in SIMPLE.

SIP method	SIMPLE function
REGISTER	Login/logout, enables the user to be reached by other user
INVITE	Initiating instant messaging and group messaging sessions.
REFER	Alternative method for inviting users into ongoing group messaging sessions
BYE	Terminating sessions
SUBSCRIBE	Subscribing to the presence information of other users, watcher information, group change information etc.
NOTIFY	Notifies subscribers of particular events, e.g. changes to presence information
UPDATE	Updating presence information
MESSAGE	Sending of a pager mode instant message

Table 5.3 SIMPLE usage of SIP methods

Chapter-6

RESULTS AND DISCUSSIONS

6.1 INTRODUCTION

This section describes the solutions used during the implementation of the CSP protocol. The tools and libraries utilized in the implementation are discussed. Then comparative study between two protocols used to implement instant messaging &presence services is presented.

6.2 IMPLEMENTATION

Implementation of instant messaging and presence services includes the programming language used to make the code and tools used for running that code. It also contains output responses in the form of testing logs.

6.2.1 Programming Language

Both the server and the client library were written in the C++ programming language. An alternative object-oriented language would have been the Java programming language. As C++ code generally is more efficient and takes less space than Java code, it was considered more suitable than Java especially for mobile devices, which are one possible environment of the CSP client. Furthermore, since there would be quite an amount of shared code between the client library and the server it was sensible to implement both using the same programming language.

6.2.2 Tools

The program code and binaries of the CSP protocol implementations were produced using the Microsoft Visual Studio 6.0 application development suite. However, no Windows specific features were utilized in order to make the code compliable for as many platforms as possible according to the requirements.

6.3 RESULTS

Output results are attached here in the form of testing logs

6.3.1 Logs for login logout

2006-04-25
11:45:19:601,User=mobile,Action=Login,IP=172.21.111.124,SessionID=49E666F6.000
00000.mobile,SessionType=IMPS
2006-04-25
11:45:45:101,User=mobile,Action=Logout,IP=172.21.111.124,SessionID=49E666F6.00
000000.mobile,SessionType=IMPS
11:24:40:539 [0] main: 42 Days remaining in trial version
11:45:18:257 [5] ThreadProcessConnection: Processing connection from 172.21.111.124...
11:45:18:257 [5] ThreadProcessConnection: Packet Length is 321 bytes
11:45:18:257 [5] ThreadProcessConnection: 50 4F 53 54 20 68 74 74 70 3A 2F 2F 31 37 32 2E POST http://172.
11:45:18:257 [5] ThreadProcessConnection: 32 31 2E 31 31 31 2E 31 32 34 3A 38 30 38 30 2F 21.111.124:8080/
11:45:18:257 [5] ThreadProcessConnection: 20 48 54 54 50 2F 31 2E 31 0D 0A 48 6F 73 74 3A HTTP/1.1 Host:
11:45:18:257 [5] ThreadProcessConnection: 20 31 37 32 2E 32 31 2E 31 31 31 2E 31 32 34 3A 172.21.111.124:
11:45:18:257 [5] ThreadProcessConnection: 38 30 38 30 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 8080 Content-Le
11:45:18:257 [5] ThreadProcessConnection: 6E 67 74 68 3A 20 31 32 38 0D 0A 43 6F 6E 74 65 ngth: 128 Conte
11:45:18:257 [5] ThreadProcessConnection: 6E 74 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 nt-Type: applica
11:45:18:257 [5] ThreadProcessConnection: 74 69 6F 6E 2F 76 6E 64 2E 77 76 2E 63 73 70 2E tion/vnd.wv.csp.
11:45:18:257 [5] ThreadProcessConnection: 77 62 78 6D 6C 0D 0A 43 6F 6E 6E 65 63 74 69 6F wbxml Connectio

11:45:18:257 [5] ThreadProcessConnection: 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 58 n: Keep-Alive X

11:45:18:257 [5] ThreadProcessConnection: 2D 57 41 50 2D 43 6C 69 65 6E 74 2D 49 50 3A 20 -WAP-Client-IP:

11:45:18:257 [5] ThreadProcessConnection: 31 39 32 2E 31 36 38 2E 31 30 2E 33 31 0D 0A 0D 192.168.10.31

11:45:18:257 [5] ThreadProcessConnection: 0A 03 10 6A 04 31 2E 31 00 C9 05 83 00 01 6D 6E j 1.1 mn

11:45:18:257 [5] ThreadProcessConnection: 70 80 19 01 01 72 74 76 80 20 01 75 03 6E 6F 6B p rtv u nok

11:45:18:257 [5] ThreadProcessConnection: 31 00 01 01 F3 07 83 00 01 00 01 5D 00 00 7A 03 1] z

11:45:18:257 [5] ThreadProcessConnection: 6D 6F 62 69 6C 65 00 01 4A 77 03 57 56 3A 49 4D mobile Jw WV:IM

11:45:18:257 [5] ThreadProcessConnection: 50 45 43 30 31 24 30 30 30 30 31 40 4E 4F 4B 2E PEC01$00001@NOK.

11:45:18:257 [5] ThreadProcessConnection: 53 36 30 00 01 01 00 01 4F 03 4D 44 35 00 01 72 S60 O MD5 r

11:45:18:257 [5] ThreadProcessConnection: C3 02 0E 10 01 70 03 77 76 3A 6E 6F 6B 69 61 2E p wv:nokia.

11:45:18:257 [5] ThreadProcessConnection: 31 33 34 37 30 36 36 37 37 30 00 01 01 01 01 01 1347066770

11:45:18:257 [5] ThreadProcessConnection: 01

11:45:18:257 [5] WVDecodedContent: <WV-CSP-Message xmlns="http://www.wireless-village.org/CSP1.1">

11:45:18:257 [5] WVDecodedContent: <Session>

11:45:18:257 [5] WVDecodedContent: <SessionDescriptor>

11:45:18:257 [5] WVDecodedContent: <SessionType>

11:45:18:257 [5] WVDecodedContent: Outband

11:45:18:257 [5] WVDecodedContent: </SessionType>

11:45:18:257 [5] WVDecodedContent: </SessionDescriptor>

11:45:18:257 [5] WVDecodedContent: <Transaction>
11:45:18:257 [5] WVDecodedContent: <TransactionDescriptor>
11:45:18:257 [5] WVDecodedContent: <TransactionMode>
11:45:18:257 [5] WVDecodedContent: Request
11:45:18:257 [5] WVDecodedContent: </TransactionMode>
11:45:18:257 [5] WVDecodedContent: <TransactionID>
11:45:18:257 [5] WVDecodedContent: nok1
11:45:18:257 [5] WVDecodedContent: </TransactionID>
11:45:18:257 [5] WVDecodedContent: </TransactionDescriptor>
11:45:18:257 [5] WVDecodedContent: <TransactionContent xmlns="http://www.wireless-village.org/TRC1.1">
11:45:18:257 [5] WVDecodedContent: <Login-Request>
11:45:18:257 [5] WVDecodedContent: <UserID>
11:45:18:257 [5] WVDecodedContent: mobile
11:45:18:257 [5] WVDecodedContent: </UserID>
11:45:18:257 [5] WVDecodedContent: <ClientID>
11:45:18:257 [5] WVDecodedContent: <URL>
11:45:18:257 [5] WVDecodedContent: WV:IMPEC01$00001@NOK.S60
11:45:18:257 [5] WVDecodedContent: </URL>
11:45:18:257 [5] WVDecodedContent: </ClientID>
11:45:18:257 [5] WVDecodedContent: <DigestSchema>
11:45:18:257 [5] WVDecodedContent: MD5
11:45:18:257 [5] WVDecodedContent: </DigestSchema>
11:45:18:257 [5] WVDecodedContent: <TimeToLive>
11:45:18:257 [5] WVDecodedContent: 3600
11:45:18:257 [5] WVDecodedContent: </TimeToLive>
11:45:18:257 [5] WVDecodedContent: <SessionCookie>
11:45:18:257 [5] WVDecodedContent: wv:nokia.1347066770
11:45:18:257 [5] WVDecodedContent: </SessionCookie>
11:45:18:257 [5] WVDecodedContent: </Login-Request>
11:45:18:257 [5] WVDecodedContent: </TransactionContent>

11:45:18:257 [5] WVDecodedContent: </Transaction>

11:45:18:257 [5] WVDecodedContent: </Session>

11:45:18:257 [5] WVDecodedContent: </WV-CSP-Message>

11:45:18:257 [5] ThreadProcessConnection: Got Login-Request transaction

11:45:18:257 [5] ThreadProcessConnection: <?xml version="1.0" encoding="UTF-8"?>

<WV-CSP-Message xmlns="http://www.wireless-village.org/CSP1.1">

<Session>

<SessionDescriptor>

<SessionType>

Outband

</SessionType>

</SessionDescriptor>

<Transaction>

<TransactionDescriptor>

<TransactionMode>

Response

</TransactionMode>

<TransactionID>

nok1

</TransactionID>

</TransactionDescriptor>

<TransactionContent xmlns="http://www.wireless-village.org/TRC1.1">

<Login-Response>

<ClientID>

<URL>

WV:IMPEC01$00001@NOK.S60

</URL>

</ClientID>

<Result>

<Code>

401

</Code>
<Description>
Please complete authentication challenge
</Description>
</Result>
<Nonce>
fda8bf106d23b5aad44a8c5bbd2ff219
</Nonce>
<DigestSchema>
MD5
</DigestSchema>
<CapabilityRequest>
F
</CapabilityRequest>
</Login-Response>
</TransactionContent>
</Transaction>
</Session>
</WV-CSP-Message>

11:45:18:273 [5] WVSendResponse: Packet Length is 189 bytes
11:45:18:273 [5] WVSendResponse: 03 10 6A 06 78 6D 6C 6E 73 00 C9 05 03 31 2E 31 j xmlns 1.1
11:45:18:273 [5] WVSendResponse: 00 01 6D 6E 70 80 19 01 01 72 74 76 80 21 01 75 mnp rtv ! u
11:45:18:273 [5] WVSendResponse: 03 6E 6F 6B 31 00 01 01 F3 07 03 31 2E 31 00 01 nok1 1.1
11:45:18:273 [5] WVSendResponse: 00 01 5E 00 00 4A 77 03 57 56 3A 49 4D 50 45 43 ^ Jw WV:IMPEC
11:45:18:273 [5] WVSendResponse: 30 31 24 30 30 30 30 31 40 4E 4F 4B 2E 53 36 30 01$00001@NOK.S60

11:45:18:273 [5] WVSendResponse: 00 01 01 6A 4B C3 02 01 91 01 52 03 50 6C 65 61 jK R Plea

11:45:18:273 [5] WVSendResponse: 73 65 20 63 6F 6D 70 6C 65 74 65 20 61 75 74 68 se complete auth

11:45:18:273 [5] WVSendResponse: 65 6E 74 69 63 61 74 69 6F 6E 20 63 68 61 6C 6C entication chall

11:45:18:273 [5] WVSendResponse: 65 6E 67 65 00 01 01 00 01 60 03 66 64 61 38 62 enge ` fda8b

11:45:18:273 [5] WVSendResponse: 66 31 30 36 64 32 33 62 35 61 61 64 34 34 61 38 f106d23b5aad44a8

11:45:18:273 [5] WVSendResponse: 63 35 62 62 64 32 66 66 32 31 39 00 01 4F 03 4D c5bbd2ff219 O M

11:45:18:273 [5] WVSendResponse: 44 35 00 01 4B 80 0B 01 01 01 01 01 01 D5 K

11:45:19:585 [5] ThreadProcessConnection: Packet Length is 342 bytes

11:45:19:585 [5] ThreadProcessConnection: 50 4F 53 54 20 68 74 74 70 3A 2F 2F 31 37 32 2E POST http://172.

11:45:19:585 [5] ThreadProcessConnection: 32 31 2E 31 31 31 2E 31 32 34 3A 38 30 38 30 2F 21.111.124:8080/

11:45:19:585 [5] ThreadProcessConnection: 20 48 54 54 50 2F 31 2E 31 0D 0A 48 6F 73 74 3A HTTP/1.1 Host:

11:45:19:585 [5] ThreadProcessConnection: 20 31 37 32 2E 32 31 2E 31 31 31 2E 31 32 34 3A 172.21.111.124:

11:45:19:585 [5] ThreadProcessConnection: 38 30 38 30 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 8080 Content-Le

11:45:19:585 [5] ThreadProcessConnection: 6E 67 74 68 3A 20 31 34 39 0D 0A 43 6F 6E 74 65 ngth: 149 Conte

11:45:19:585 [5] ThreadProcessConnection: 6E 74 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 nt-Type: applica

11:45:19:585 [5] ThreadProcessConnection: 74 69 6F 6E 2F 76 6E 64 2E 77 76 2E 63 73 70 2E tion/vnd.wv.csp.

11:45:19:585 [5] ThreadProcessConnection: 77 62 78 6D 6C 0D 0A 43 6F 6E 6E 65 63 74 69 6F wbxml Connectio

11:45:19:585 [5] ThreadProcessConnection: 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 58 n: Keep-Alive X

11:45:19:585 [5] ThreadProcessConnection: 2D 57 41 50 2D 43 6C 69 65 6E 74 2D 49 50 3A 20 -WAP-Client-IP:

11:45:19:585 [5] ThreadProcessConnection: 31 39 32 2E 31 36 38 2E 31 30 2E 33 31 0D 0A 0D 192.168.10.31

11:45:19:585 [5] ThreadProcessConnection: 0A 03 10 6A 04 31 2E 31 00 C9 05 83 00 01 6D 6E j 1.1 mn

11:45:19:585 [5] ThreadProcessConnection: 70 80 19 01 01 72 74 76 80 20 01 75 03 6E 6F 6B p rtv u nok

11:45:19:585 [5] ThreadProcessConnection: 31 00 01 01 F3 07 83 00 01 00 01 5D 00 00 7A 03 1] z

11:45:19:585 [5] ThreadProcessConnection: 6D 6F 62 69 6C 65 00 01 4A 77 03 57 56 3A 49 4D mobile Jw WV:IM

11:45:19:585 [5] ThreadProcessConnection: 50 45 43 30 31 24 30 30 30 30 31 40 4E 4F 4B 2E PEC01$00001@NOK.

11:45:19:585 [5] ThreadProcessConnection: 53 36 30 00 01 01 00 01 4E 03 72 5A 6A 4A 6E 78 S60 N rZjJnx

11:45:19:585 [5] ThreadProcessConnection: 6A 66 6B 55 35 79 32 30 6A 46 36 35 4B 46 49 51 jfkU5y20jF65KFIQ

11:45:19:585 [5] ThreadProcessConnection: 3D 3D 00 01 72 C3 02 0E 10 01 70 03 77 76 3A 6E == r p wv:n

11:45:19:585 [5] ThreadProcessConnection: 6F 6B 69 61 2E 31 33 34 37 30 36 36 37 37 30 00 okia.1347066770

11:45:19:585 [5] ThreadProcessConnection: 01 01 01 01 01 01

11:45:19:585 [5] WVDecodedContent: <WV-CSP-Message xmlns="http://www.wireless-village.org/CSP1.1">

11:45:19:585 [5] WVDecodedContent: <Session>

11:45:19:585 [5] WVDecodedContent: <SessionDescriptor>

11:45:19:585 [5] WVDecodedContent: <SessionType>
11:45:19:585 [5] WVDecodedContent: Outband
11:45:19:585 [5] WVDecodedContent: </SessionType>
11:45:19:585 [5] WVDecodedContent: </SessionDescriptor>
11:45:19:585 [5] WVDecodedContent: <Transaction>
11:45:19:585 [5] WVDecodedContent: <TransactionDescriptor>
11:45:19:585 [5] WVDecodedContent: <TransactionMode>
11:45:19:585 [5] WVDecodedContent: Request
11:45:19:585 [5] WVDecodedContent: </TransactionMode>
11:45:19:585 [5] WVDecodedContent: <TransactionID>
11:45:19:585 [5] WVDecodedContent: nok1
11:45:19:585 [5] WVDecodedContent: </TransactionID>
11:45:19:585 [5] WVDecodedContent: </TransactionDescriptor>
11:45:19:585 [5] WVDecodedContent: <TransactionContent xmlns="http://www.wireless-village.org/TRC1.1">
11:45:19:585 [5] WVDecodedContent: <Login-Request>
11:45:19:585 [5] WVDecodedContent: <UserID>
11:45:19:585 [5] WVDecodedContent: mobile
11:45:19:585 [5] WVDecodedContent: </UserID>
11:45:19:585 [5] WVDecodedContent: <ClientID>
11:45:19:585 [5] WVDecodedContent: <URL>
11:45:19:585 [5] WVDecodedContent: WV:IMPEC01$00001@NOK.S60
11:45:19:585 [5] WVDecodedContent: </URL>
11:45:19:585 [5] WVDecodedContent: </ClientID>
11:45:19:585 [5] WVDecodedContent: <DigestBytes>
11:45:19:585 [5] WVDecodedContent: rZjJnxjfkU5y20jF65KFIQ==
11:45:19:585 [5] WVDecodedContent: </DigestBytes>
11:45:19:585 [5] WVDecodedContent: <TimeToLive>
11:45:19:585 [5] WVDecodedContent: 3600
11:45:19:585 [5] WVDecodedContent: </TimeToLive>
11:45:19:585 [5] WVDecodedContent: <SessionCookie>

11:45:19:585 [5] WVDecodedContent: wv:nokia.1347066770

11:45:19:585 [5] WVDecodedContent: </SessionCookie>

11:45:19:585 [5] WVDecodedContent: </Login-Request>

11:45:19:585 [5] WVDecodedContent: </TransactionContent>

11:45:19:585 [5] WVDecodedContent: </Transaction>

11:45:19:585 [5] WVDecodedContent: </Session>

11:45:19:585 [5] WVDecodedContent: </WV-CSP-Message>

11:45:19:585 [5] ThreadProcessConnection: Got Login-Request transaction

11:45:19:585 [5] SetXmppUserPresence: <status>Connected to IM on a Mobile Phone</status>

11:45:19:585 [5] SetXmppUserPresence: Updating last avail time

11:45:19:601 [5] ThreadProcessConnection: <?xml version="1.0" encoding="UTF-8"?>

<WV-CSP-Message xmlns="http://www.wireless-village.org/CSP1.1">

<Session>

<SessionDescriptor>

<SessionType>

Outband

</SessionType>

</SessionDescriptor>

<Transaction>

<TransactionDescriptor>

<TransactionMode>

Response

</TransactionMode>

<TransactionID>

nok1

</TransactionID>

</TransactionDescriptor>

<TransactionContent xmlns="http://www.wireless-village.org/TRC1.1">

<Login-Response>

<ClientID>

<URL>
WV:IMPEC01$00001@NOK.S60
</URL>
</ClientID>
<Result>
<Code>
200</Code>
<Description>
Login OK</Description>
</Result>
<SessionID>49E666F6.00000000.mobile
</SessionID><KeepAliveTime>
90</KeepAliveTime>
<CapabilityRequest>
T</CapabilityRequest>
</Login-Response>
</TransactionContent>
</Transaction>
</Session>
</WV-CSP-Message>
2006-04-25 11:45:18,mobile,172.21.111.124,nok1,ToServer,Login-Request,mobile
2006-04-25 11:45:18,mobile,172.21.111.124,nok1,ToClient,Login-Response,401
2006-04-25 11:45:19,mobile,172.21.111.124,nok1,ToServer,Login-Request,mobile
2006-04-25 11:45:19,mobile,172.21.111.124,nok1,ToClient,Login-Response,200
2006-04-25 11:45:21,mobile,172.21.111.124,nok2,ToServer,ClientCapability-Request
2006-04-25 11:45:21,mobile,172.21.111.124,nok2,ToClient,ClientCapability-Response
2006-04-25 11:45:22,mobile,172.21.111.124,nok3,ToServer,Service-Request
2006-04-25 11:45:22,mobile,172.21.111.124,nok3,ToClient,Service-Response
2006-04-25 11:45:23,mobile,172.21.111.124,nok4,ToServer,GetList-Request
2006-04-25 11:45:23,mobile,172.21.111.124,nok4,ToClient,GetList-Response

2006-04-2511:45:25,mobile,172.21.111.124,nok5,ToServer,UnsubscribePresence-Request,wv:mobile/~IM1.0_friendslist
2006-04-25 11:45:25,mobile,172.21.111.124,nok5,ToClient,Status,200
2006-04-2511:45:26,mobile,172.21.111.124,nok6,ToServer,ListManage-Request,wv:mobile/~IM1.0_friendslist
2006-04-25 11:45:26,mobile,172.21.111.124,nok6,ToClient,ListManage-Response,200
2006-04-2511:45:28,mobile,172.21.111.124,nok7,ToServer,CreateAttributeList-Request,DefaultList
2006-04-25 11:45:28,mobile,172.21.111.124,nok7,ToClient,Status,200
2006-04-2511:45:29,mobile,172.21.111.124,nok8,ToServer,UpdatePresence-Request,"OnlineStatus,ClientInfo,CommCap,UserAvailability,StatusText,StatusContent"
2006-04-25 11:45:29,mobile,172.21.111.124,nok8,ToClient,Status,200
2006-04-25 11:45:30,mobile,172.21.111.124,,ToServer,Polling-Request
2006-04-25 11:45:31,mobile,172.21.111.124,nok9,ToServer,GetBlockedList-Request
2006-04-25 11:45:31,mobile,172.21.111.124,nok9,ToClient,GetBlockedList-Response
2006-04-2511:45:33,mobile,172.21.111.124,nok10,ToServer,Search-Request,GROUP_USER_ID_OWNER=mobile
2006-04-2511:45:33,mobile,172.21.111.124,nok10,ToClient,Search-Response,SearchFindings=0
2006-04-2511:45:35,mobile,172.21.111.124,nok11,ToServer,SubscribePresence-Request,wv:mobile/~IM1.0_friendslist
2006-04-25 11:45:35,mobile,172.21.111.124,nok11,ToClient,Status,200
2006-04-2511:45:35,mobile,172.21.111.124,wvnow4419a6b5,ToClient,
PresenceNotification-Request
2006-04-25 11:45:36,mobile,172.21.111.124,,ToServer,Polling-Request
2006-04-25 11:45:36,mobile,172.21.111.124,nok12,ToServer,StopSearch-Request
2006-04-25 11:45:36,mobile,172.21.111.124,nok12,ToClient,Status,200
2006-04-25 11:45:36,mobile,172.21.111.124,,ToServer,Polling-Request
2006-04-25 11:45:37,mobile,172.21.111.124,wvnow4419a6b5,ToServer,Status,200
2006-04-25 11:45:38,mobile,172.21.111.124,,ToServer,Polling-Request

2006-04-2511:45:43,mobile,172.21.111.124,nok13,ToServer,UpdatePresence-Request,CommCap

2006-04-25 11:45:43,mobile,172.21.111.124,nok13,ToClient,Status,200

2006-04-25 11:45:45,mobile,172.21.111.124,nok14,ToServer,Logout-Request

2006-04-25 11:45:45,mobile,172.21.111.124,nok14,ToClient, Disconnect,200

6.4 COMPARISON OF IMPS & SIMPLE

As IMPS is specifically defined for usage in mobile environments and the IP Multimedia Subsystem (IMS) brings SIMPLE to forthcoming 3G networks, these two services are the top alternatives for mobile instant messaging. IMPS is a more mature service than SIMPLE. SIMPLE has not yet reached standard status and parts of the service are still ongoing work. However, the main elements of SIMPLE are ready and the service is adopted as an IETF standard in 2005.

The functionality of both services is very similar from a user's point of view, but the techniques used to provide the functionality differ considerably between the services. IMPS is based on a rather simple architecture where all client communication passes through servers. SIMPLE on the other hand is a fairly complex solution, which relies on SIP for much of its functionality but other protocols such as XCAP and MSRP are also utilized. Both services utilize techniques for optimizing the performance in mobile environments. Overall, SIMPLE is slightly more efficient than IMPS when it comes to bandwidth usage and delays. Performance-wise, the most notable flaw in SIMPLE is the inability to traverse proxies. This affects the applicability of the service since a global service cannot be created. IMPS is able to handle proxy traversal without problems. SIMPLE includes mechanisms for providing the service with relatively strong security. Due to the complexity of the SIMPLE architecture, applying an even level of security throughout a network requires a great deal of cooperation between network administrators.

IMPS provides sufficient security only between the client and its local server, end-to-end security can not be requested by clients and is therefore not guaranteed in all networks. Since the IMPS protocols are completely based on XML and function on top of several different transport bindings, IMPS qualifies as an extensible and flexible solution. SIMPLE also utilizes the XML format, but the tight coupling with the SIP protocol reduces flexibility to an extent.

Tables 6.1 and 6.2 list the main advantages and disadvantages of the compared services.

Advantages	**Disadvantages**
Simple architecture	Security
Scalability	Lack of charging protocol
Extensibility	
Proxy reversal	

Table 6.1: Advantages and disadvantages of IMPS

Advantages	**Disadvantages**
Scalability	Complex architecture
Interoperability	Proxy reversal
Security	
Efficiency	

Table 6.2: Advantages and disadvantages of SIMPLE

Chapter-7

CONCLUSIONS & FUTURE SCOPE

7.1 CONCLUSION

Instant messaging services have enjoyed a constant growth ever since their introduction. Real-time messages and presence information are the pieces of technology that makes instant messaging different from previous communication services. However, the success of instant messaging is not based on technical differences only; also the methods and concepts used in instant messaging clients, such as popup windows and buddy lists, have contributed to the birth of a completely new type of communication. This thesis has summarized the main work ongoing in the IETF SIMPLE. The aim of present work is to compare SIP protocol with Instant Messaging and Presence functionalities with IMPS respecting the requirements. Present work defines a general framework for Instant Messaging and Presence, and SIMPLE builds a SIP-based solution on top of the IMPP framework.

As IMPS is specifically defined for usage in mobile environments and the IP Multimedia Subsystem (IMS) brings SIMPLE to forthcoming 3G networks, these two services are the top alternatives for mobile instant messaging. IMPS is a more mature service than SIMPLE. SIMPLE has not yet reached standard status and parts of the service are still ongoing work. However, the main elements of SIMPLE are ready and the service should be adopted as an IETF standard in 2005. The functionality of both services is very similar from a user's point of view, but the techniques used to provide the functionality differ considerably between the services. IMPS is based on a rather simple architecture where all client communication passes through servers. SIMPLE on the other hand is a fairly complex solution that relies on SIP for much of its functionality but other protocols such as XCAP and MSRP are also utilized. Both services utilize techniques for optimizing the performance in mobile environments. Overall, SIMPLE is slightly more efficient than IMPS when it comes to bandwidth usage and delays. Performance-wise, the

most notable flaw in SIMPLE is the inability to traverse proxies. This affects the applicability of the service since a global service cannot be created. IMPS is able to handle proxy traversal without problems.

SIMPLE includes mechanisms for providing the service with relatively strong security. Due to the complexity of the SIMPLE architecture, applying an even level of security throughout a network requires a great deal of cooperation between network administrators. IMPS provides sufficient security only between the client and its local server, end-to-end security can not be requested by clients and is therefore not guaranteed in all networks. Since the IMPS protocols are completely based on XML and function on top of several different transport bindings, IMPS qualifies as an extensible and flexible solution.

Finally, both services are built based on the requirements formulated by the IMPP working group, thus enabling good interoperability with other instant messaging systems. However, IMPS does not support the PIDF presence format, whereas SIMPLE is fully compliant with IMPP.

7.2 FUTURE WORK

It appears that, if the security problem of spoofing can be solved, SIMPLE would be a viable and easy to implement approach to P2P based IM. The leveraging of existing clients will allow this to be more easily implemented. Using SIP/SIMPLE as the IM protocol will allow for rapid integration of multimedia sessions. As the functionality of the IMPS is already being defined but many parts of the SIMPLE specifications are not finalized, updates and improvements will be made to both the compared services in the future. These changes can affect the validity of the results of the comparison. Therefore, the comparison should be kept up to date as the services evolve. Secondly IMS is upcoming NGN architecture, which uses SIMPLE as its standard for instant messaging and presence. So more work is to be done to remove complexities in implementation. The problem of spoofing in SIMPLE is related with imitation of a principal by another principal. Authentication rules are intended to deal with spoofing. This is a matter of future study.

LIST OF PUBLICATIONS

- Presented a paper on **"WIRELESS SENSOR NETWORKS- A REVIEW"** in **NATIONAL CONFERENCE ON SENSORS** at Thapar Institute Of Engineering & Technology, Patiala in Nov2005.

- Presented a paper on **"A REVIEW OF PROBLEMS IN WIRLESS SENSOR NETWORKS"** in **NATIONAL CONFERENCE ON SENSORS** at Thapar Institute Of Engineering & Technology, Patiala in Nov2005.

REFERENCES

[1] 3G Americas – IP Multimedia Subsystem (IMS) overview and applications white paper

[2]Gonzalo Camarillo and Miguel a. Garcia-Martin – The 3G IP Multimedia Subsystem, Merging the Internet and the cellular worlds (2004)

[3]Handley, M., et al., "SIP: Session Initiation Protocol," RFC 2543, 1999, Section 6.

[4]Fielding, R., et al., "Hypertext Transfer Protocol — HTTP/1.1," RFC 2068, June 1999.

SIP: Session Initiation Protocol, IETF RFC 3261, Rosenberg, et. al

[5] 3GPP TS 24.228 Signalling flows for the IP multimedia call control based on SIP and SDP (Release5)

[6] 3GPP TS 23.002 Network architecture (Release 5)

[7] 3GPP TS 23.228 IP Multimedia Subsystem (IMS) (Release 5)

[8] 3GPP TS 33.203 Access security for IP based services (Release 5)

[9] 3GPP TS 24.229: IP Multimedia Call Control Protocol based on SIP and SDP; Stage3 (Release 5)

[10] Signalling Compression (SigComp), IETF RFC 3320, Price, et. al

[11] Open Mobile Alliance. System Architecture Model, Candidate Version 1.2, May 2004.

[12] B. Campbell, R. Mahy, and C. Jennings. The Message Session Relay Protocol. Internet draft, Internet Engineering Task Force, August 2004.Work in progress, Available at: http://www.ietf.org/internet-drafts/draft-ietf-simple-message-sessions-08.txt [referred to 30.8.2004].

[13] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle. Session Initiation Protocol (SIP) Extension for Instant Messaging (RFC 3428). The Internet Society, December 2002.

[14] C. Jennings and R. Mahy. Relay Extensions for Message Sessions Relay Protocol (MSRP). Internet draft, Internet Engineering Task Force, July 2004. Work in progress, Available at: http://www.ietf.org/internet-drafts/draft-ietf-simple-msrp-relays-01.txt [referred to 30.8.2004].

[15] J. Rosenberg. A Presence Event Package for the Session Initiation Protocol (SIP) (RFC 3856). The Internet Society, August 2004.

[16] J. Rosenberg. The Extensible Markup Language (XML) Configuration Access Protocol (XCAP). Internet draft, Internet Engineering Task Force, July 2004. Work in progress, Available at: http://www.ietf.org/internet-drafts/draft-ietf-simple-xcap-03.txt [referred to 30.8.2004].

[17]M. Day, J. Rosenberg, and H. Sugano. RFC 2778: "A Model for Presence and Instant Messaging," IETF, February 2000.

[18]M.T. Rose, G. Klyne, and D.H. Crocker. The IMXP Presence Service, IETF Internet draft, March 2001, work in progress.

[19]F. Mazzoldi, A. Diacakis, S. Fujimoto, G. Hudson, J.D. Ramsdell, and H. Sugano. Presence and Instant Messaging Protocol (PRIM), IETF Internet draft, September 2000, work in progress.

[20]D. Crocker, D. Athanasios, C. Huitema, G. Klyne, F. Mazzoldi, M. Rose, J. Rosenberg, R. Sparks, and H. Sugano. A Framework for Moving IMPP Forward, IETF Internet draft, August 2000, work in progress.

[21]D. Crocker, D. Athanasios, C. Huitema, G. Klyne, F. Mazzoldi, M. Rose, J. Rosenberg, R. Sparks, and H. Sugano. "A Common Profile for Instant Messaging (CPIM)," IETF Internet draft, November 2000, work in progress.

[22]D. Crocker. RFC 822: "Standard for the Format of ARPA Internet Text Messages," August 1982.

[23]The Instant Messaging and Presence Protocol (IMPP) Working Group of the IETF http://ietf.org/html.charters/impp-charter.html.

[24]J. Rosenberg, D. Willis, R. Sparks, B. Campbell, H. Schulzrinne, J. Lennox, B. Aboba, C. Huitema, and D. Gurle. A Protocol for Presence Based in SIP, IETF Internet draft, June 2000, work in progress.