

Chapter – 7

Study & Modeling Instant Messaging and Presence over IMS

- 7.1 Introduction
 - 7.1.1 Definitions
- 7.2 Concepts and Model for IMPS
 - 7.2.1 Presence Service
 - 7.2.2 Instant Message Service
 - 7.2.3 Differences from Other Communication Systems Model
 - 7.2.4 History of Messaging over Wireless Network
 - 7.2.5 Differences in Comparison to Instant Messaging in Fixed / wireless Networks
 - 7.2.5.1 Device Limitations
 - 7.2.5.2 Network Limitations
 - 7.2.5.3 Mobility
 - 7.2.5.4 Other Factors
 - 7.2.6 Standards and Protocols
 - 7.2.6.1 xMS (SMS, EMS, MMS)
 - 7.2.6.2 Short Message Service (SMS)
 - 7.2.6.3 Enhanced Messaging Service (EMS)
 - 7.2.6.4 Multimedia Messaging Service (MMS)
 - 7.2.6.5 IMPS
- 7.3 Entities of Presence & Instant Messaging
- 7.4 Study of IMPS and SIMPLE Protocols
 - 7.4.1 Functionality
 - 7.4.2 Evaluate Both Technologies
 - 7.4.2.1 Architecture
 - 7.4.2.2 Protocols
 - 7.4.3 Culmination for further implementation of Research
- 7.5 The Common Profile for Instant Messaging (IM)
- 7.6 Example of Presence Service
- 7.7 Example of Instant Message Service
- 7.8 Performance evaluation of IMPS and SIMPLE
 - 7.8.1 Bandwidth Usage

7.8.2 Delays

7.8.3 Coverage Loss

7.8.4 Proxy Traversal

7.8.5 Scalability

7.9 IMPLEMENTATION

7.10 RESULTS

7.11 Comparison of IMPS & SIMPLE Performance

7.12 Analysis of presence server traffic load

7.13 Mathematical Analysis

7.14 Dimension IMPS over a PoC service

7.14.1 Proposed Access priority Model

7.14.2 Proposed Timer

7.14.3 Proposed Model to Optimize Simultaneous Sessions

7.15 Summary

Chapter – 7

Study & Modeling Instant Messaging and Presence over IMS

7.1 Introduction

This chapter, cover study over Instant Messaging and Presence service over IMS Network. During our research work had study all existing models of IMPS with respective architecture. And over our own IMS-Client we deployed IMPS service. With this chapter we discuss brief of Instant Messaging services protocols, standard and its implementation over our IMS-Client, we also have shows how they can be implemented using SIP in IP multimedia subsystems with results discussion.

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 [140]. 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. Although first considered a toy for teenagers, the value of instant messaging to enterprises, e.g. in form of cost savings and increase in efficiency, has been recognized in recent years.

7.1.1 Definitions

Several different interpretations of the term 'instant messaging' exist. This section clarifies the meaning of the term within this thesis. IEC (International Engineering Consortium) provides the following definition: "Instant messaging (IM) is an Internet protocol (IP)-based application that provides convenient communication between people using a variety of different device types" [141]. Campbell et al. [142] defines instant messaging as "the exchange of content between a set of participants in near real time".

Webopedia.com [143] states that instant messaging is "a type of communications service that enables you to create a kind of private chat room with another individual in order to communicate in real time over the Internet, analogous to a telephone conversation but using text-based, not voice-based, communication. Typically, the instant messaging system alerts you whenever somebody on your private list is online".

According to another online IT-specific encyclopedia, Whatis.com [144], instant messaging is "the ability to easily see whether a chosen friend or co-worker is connected to the Internet and, if they are, to exchange messages with them". Finally, Kohda et al. [145] refers to instant messaging as "buddy list applications, which consist of two orthogonal services, presence and short messaging". The IEC definition is rather vague and applies to almost all types of messaging. Campbell et al. recognizes the fact that instant messages are delivered to the recipient in real time.

The remaining definitions all include presence as a part of instant messaging, thus exposing the dual nature of the term. Instant messaging is often perceived as a service consisting of both presence and sending of instant messages. However, the term instant messaging is frequently used also when referring to the sending of instant messages. In this thesis the former definition will mainly be used. If the term “instant messaging” is referring to the sending of instant messages it will be made clear by context.

Definition of Instant Messaging

“Instant messaging is a type of communication service providing users with two elements; presence information and real-time messaging. As it is an essential part of instant messaging it is necessary to provide a definition for presence as well.”

Webopedia.com [143] defines presence as “the ability to detect whether other users are online and whether they are available”. Day et al. [146] provides the following definition: “Presence is a means for finding, retrieving, and subscribing to changes in the presence information (e.g. “online” or “offline”) of other users”.

For the purposes of this thesis, the latter definition is suitable. It should be stressed that presence information is not restricted to online status only; other attributes such as mood, location and language might just as well be included. Also note that instant messaging is merely one application of presence; other technologies, e.g. VoIP (Voice over IP), provide presence services as well.

Definition of Presence

“Presence is a means for finding, retrieving, and subscribing to changes in the presence information of other users.”

7.2 Concepts and Model for IMPS

RFC (Request For Comment) 2778 [149], produced by the IETF IMPP working group, defines a common vocabulary and presents an abstract model for instant messaging. This model is used as such by all current solutions, both proprietary and open standards.

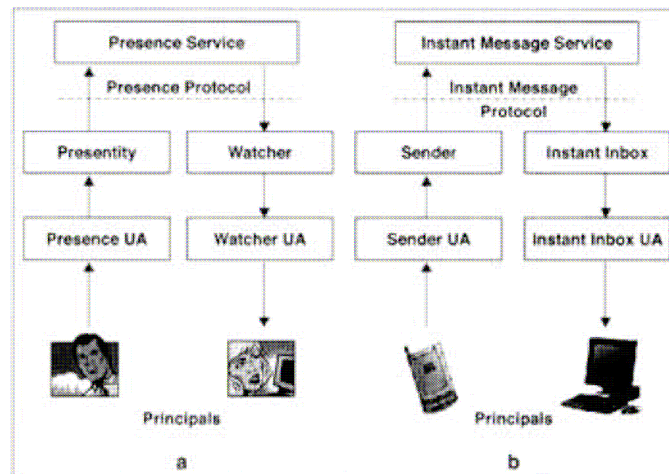


Figure 7.1 Instant Messaging and Presence on User Equipment

7.2.1 Presence Service

Figure 7.2 displays an overview of a presence service. A presence service has two distinct types of clients: 'presentities' and 'watchers'. Presentities provide presence information to the presence service, while watchers request presence information about presentities from the presence service. Naturally, the same application can act both as presentity and as a watcher.

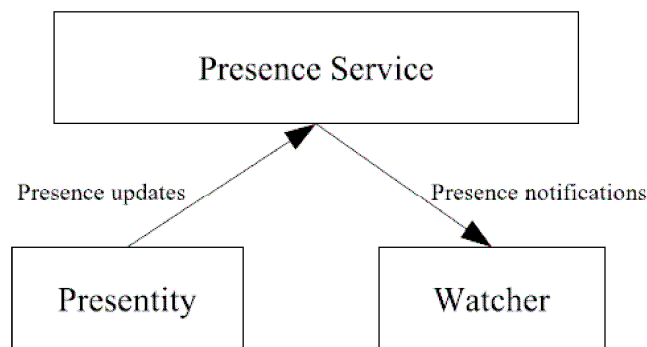


Figure 7.2: Overview of a presence service

The watchers are classified as well; there are 'subscribers', 'fetchers' and 'pollers' (see Figure 7.3). A subscriber is a watcher that has subscribed to the presence information of presentity. The presence service keeps track of the subscriptions and sends a notification to the subscriber whenever the presence information of the subscribed presentity changes. A fetcher requests the presence service for presence information about presentity. The presence service does not send notifications to fetchers, presence information is only sent upon the request of the fetcher. A poller is a special kind of a fetcher that polls the presence service for presence information about presentity on a regular basis.

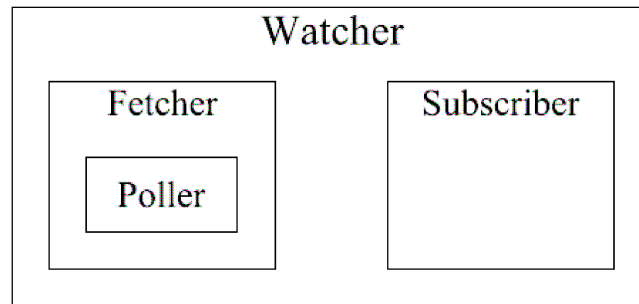


Figure 7.3: Different kinds of watchers

7.2.2 Instant Message Service

Equally to the presence service, the instant message service also has two kinds of clients: 'senders' and 'instant inboxes' (see Figure 7.4). Senders are the source of instant messages to be delivered by the instant message service. An instant inbox is a container for instant messages that are to be read by the owner of the inbox. The instant message service accepts instant messages from senders and attempts to deliver them to the instant inboxes, to which they are addressed. Typically, the instant inbox resides in the client application, although it is not required by the IMPP definition.

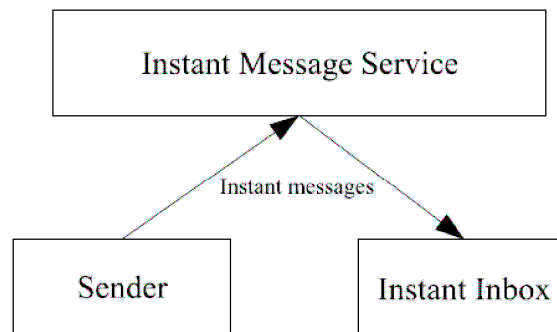


Figure 7.4: Overview of an instant message service

7.2.3 Differences from Other Communication Systems Model

Presence, as such, is clearly not part of any major established communication systems of today, but also the method for delivering messages differs considerably from the methods currently used.

Most of the currently used messaging systems are based on a paradigm called 'store-and-forward'. When sending a message over a network using this paradigm, each message transfer agent stores and forwards the message to the next agent. In case the next agent cannot be reached, delivery can be reattempted later on. The store of the message transfer agent closest to the user agent of the recipient usually serves as an inbox to the user agent. The user agent fetches the messages from the inbox when it is online. Even though it is possible for messages to arrive real-time, 'store-and-forward' is viewed as non-real-time delivery system since all delivered messages do not

arrive in real-time. Examples of messaging systems using 'store-and-forward' include traditional email, SMS (Short Message Service) and MMS (Multimedia Messaging Service).

For instant messaging systems, intermediate elements along the path from the sender to the recipient merely relay the instant message to the next element without storing it. If the recipient cannot be reached, e.g. when the user is not online, then the instant message is normally discarded. Some instant messaging systems provide storage for messages sent to offline users, but this is a distinct service and not part of instant messaging as such.

Sending instant messages from peer to peer is also possible using a few Instant Messaging systems (e.g. ICQ and SIMPLE). These systems provide the sender with the address of the recipient, using which the instant messages can be sent directly to the recipient without any server intervention.

Usage

Voice calls and email are by far the two most popular communication systems of today [150]. Table 7.1 shows some usage related properties of instant messaging compared with the corresponding properties of voice calls and email. Instant messaging can be seen as some kind of combination of features from voice calls and email. Like voice calls instant messaging provides real-time communication but at a price comparable to email. Of course, instant messaging is not free; deploying an instant messaging solution in an enterprise requires noticeable investments.

However, after deployment the cost of usage is significantly lower for instant messaging than for voice calls, especially for international communication. When it comes to archiving, email systems provide the most sophisticated solutions. Presence information is one of the greatest benefits for users of instant messaging. Surveys show that 40-60% of business related phone calls fail due to the callee being busy or absent [151]. Presence information practically eliminates these phenomena for instant messaging, assuming that users keep their presence information up-to-date.

Table 7.1: Comparison of popular messaging systems

Messaging System	Real-time	Cost	Presence	Archiving
Voice call	Yes	High	Partly, the call is either established or not	No
Email	No	Low	No, the sender does not know whether the recipient is available or not.	Yes
Instant Messaging	Yes	Low	Yes, the status of the recipient is known by the system at all times.	Depends on the IM client. Must often be done explicitly by the user.

Table 7.2 shows how initiating a business related line of communication using different messaging systems affects the productivity of parties involved. Voice calls often interrupt the work of the callee, and on the contrary, emailing is more prone to suspend caller's work for a while. Instant messaging is more like voice calls in this aspect, but presence information provides a means to prevent unnecessary interruptions. Instant messaging can also reduce productivity if not deployed carefully within an organization. Gossiping and constant interruptions are some aspects that can decrease productivity. The productivity aspect of instant messaging is discussed in more detail in [152] and [153].

Table 7.2: The influence of messaging systems on productivity

Messaging System	Caller	Callee
Voice call	Generally no decrease in productivity. If the callee is reached, a direct answer can be received.	Often decreases productivity as the interrupted callee must drop its current work in order to answer the call.
Email	Decrease in productivity while waiting for a reply.	Affects productivity only slightly since the callee can choose when to answer the email.
Instant Messaging	Typically no productivity reduction. The presence service allows the caller to select currently available callee.	Reduces productivity upon message reception, but the callee can use the presence service to control the arrival of messages.

A niche for instant messaging as it is more suitable for particular tasks than the current systems. However, instant messaging will not eliminate any of the current systems. Users will choose which system to use based on the type of communication. Instant messaging is suitable for quick real-time conversations, email is convenient for errands that do not require an immediate response and voice calls are often preferred for e.g. business negotiations as the risk of misunderstandings is reduced.

7.2.4 History of Messaging over Wireless Network

As wireless networks only have existed for a few decades, the history of mobile messaging is short and the history of mobile instant messaging even shorter. Mobile messaging took off in the late 1990's when SMS messaging was made available to GSM (Global System for Mobile communications) subscribers. In the following years, SMS usage kept rising rapidly as can be seen from Figure 7.5

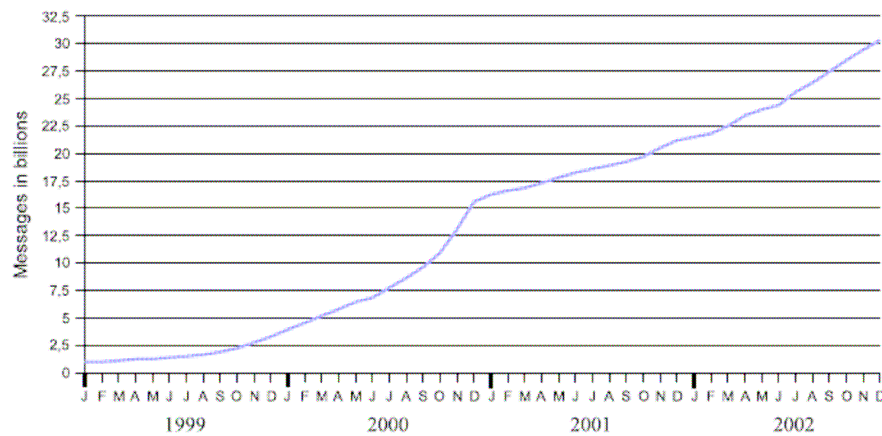


Figure 7.5: Worldwide monthly SMS traffic. Source: EMC Research

The success of SMS got the operators of mobile networks to realize the value of mobile messaging; considerable revenues can be gained using only a fraction of the available bandwidth. Therefore, new messaging services similar to SMS but with richer features, such as EMS (Enhanced Messaging Service) and MMS, have been designed and deployed lately.

The most successful messaging system of the Internet, email, has also been brought to mobile clients in a number of different ways. In the beginning, email in mobile clients was handled through SMS messages. Later, mobile email was made available through WAP (Wireless Application Protocol) and recently mobile devices with email clients managing email using well-known mail protocols such as IMAP (Internet Message Access Protocol) or POP (Post Office Protocol) on top of GPRS (General Packet Radio Service) have been made available.

Instant messaging solutions for mobile devices have been available since 2002, when AOL [154] and Yahoo started providing access to their instant messaging network using an SMS interface. However, these services are not available worldwide, only users of wireless carriers that have made an agreement with AOL or Yahoo can use them. The SMS-based instant messaging services are not very convenient to use, as they require the user to remember several commands and phone numbers. The recent introduction of mobile devices that allow both installation of third party software and packet switched data transfer has enabled companies to develop their own mobile client software for their proprietary protocols. For example, AOL has released an instant messaging client for mobile devices running the Symbian operating system.

Work for a non-proprietary mobile instant messaging solution commenced in 2001 when Ericsson, Nokia and Motorola teamed up to form the Wireless Village initiative (now known as IMPS) [155], a joint project established to create universal specifications for mobile instant messaging. The first release of the specifications was made available in 2002 but it was not until in the fourth quarter of 2003 that the first mobile device with support for the technology was published.

The IP Multimedia Subsystem defined for enabling multimedia communication services for 3G networks uses SIP for establishing multimedia sessions. The upcoming 3GPP Release 6 specifications include instant messaging using SIMPLE, bringing SIMPLE forth as a contender to IMPS in the mobile instant messaging world as soon as IMS is widely deployed.

7.2.5 Differences in Comparison to Instant Messaging in Fixed / wireless Networks

Transitioning from developing Internet services targeted at fixed networks to developing mobile services is not a trivial task. The mobile environment introduces several constraints on a mobile service that do not exist in traditional, fixed environments. Not only does the technology used for establishing wireless networks place limitations on a mobile service, but also mobile terminals and the behavior of users differ significantly between fixed and mobile environments. This section presents the characteristics of mobile environments that deviate from the corresponding ones of wireline environments. Even though only aspects affecting mobile instant messaging services are considered here, most of them apply to mobile services in general.

7.2.5.1 Device Limitations

Due to the requirement of portability, mobile devices have several differences in comparison to stationary computers, which should be considered when designing a mobile service. These differences are presented briefly below; in-depth studies are available in [156], [157] and [158].

A. Display Size

As mobile devices are pocket-sized for portability, it follows that displays must be relatively small to fit the devices. Due to the small display size, special emphasis must be put on user interface design for mobile applications. In addition, there are no standard display resolutions for mobile devices. There exist high-level languages allowing the target device to render the user interface using its native style, thus trading look-and-feel design for portability. However, in practice distinct versions of the user interface must often be created for different display types.

B. Input Device

Size constraints of portable devices make many traditional input devices such as keyboards and mice impossible to use with mobile devices. Instead, mobile devices come equipped with limited keyboards, navigator buttons, touch screens or pointing devices. When it comes to instant messaging, the greatest problem arising from these alternative input devices is the difficulty of typing. Typing using any of the aforementioned methods is notably slower than typing using a regular keyboard. In recent years some progress has been

made in this area; for example the T9 text input system has been introduced.

C. Memory

Memory is another aspect differing in size from wireline environments to mobile environments. Both physical memory and permanent storage memory are limited on mobile devices.

These memory limitations cannot be ignored when designing a mobile service. Due to memory constraints some handsets might e.g. be unable to receive large bulks of data, forcing designers to utilize alternative methods such as sending data in smaller chunks.

Luckily, rapid progress has been made in this area. In the last few years the magnitude of memory sizes has shifted from KB to MB. Several solutions for storing data permanently fitting into a very small space, such as CompactFlash, MultiMediaCard, Secure Digital Card and SmartMedia Card, have been put into market and are used in numerous new mobile devices.

D. Processing Power

In addition to memory limitations, mobile devices are limited also when it comes to processing power. When designing a mobile instant messaging service the restrictions of mobile device processors need to be considered especially when deciding what data formats to use. Data formats are of importance because of the processing power that is required to en/decode them for network transmission. Codecs for some formats are very efficient while others require heavy computing. In addition, some formats require more processing power to encode than to decode and vice versa. In general, codecs demanding more processing power often result in a better data compression rate and since data size is another important aspect of mobile computing, this trade-off between processing power and data size must also be taken into account when choosing data formats.

E. Battery

Batteries power practically all-mobile devices. As most batteries are rechargeable, saving batteries is often quite nonessential. Furthermore, software engineers implementing a mobile service have more influence on battery usage than the designers of the service.

A designer of a mobile service can slightly affect the usage of power by optimizing the use of the processor and data communications. Data communication hardware consumes power especially when sending or receiving data, optimizing the amount of sent data can help reducing power consumption. For techniques reducing the need for processing power, see the previous subsection.

F. Media Formats

The operating systems of mobile handsets come equipped with a limited range of natively supported media formats. When specifying a mobile service, support for a particular media type cannot be assumed. Preferably a mobile instant messaging service should provide the means for parties of a communication to find out the mutually supported formats or at least notify the sender of a message when the recipient does not support a particular media format.

7.2.5.2 Network Limitations

The characteristics of wireless network links differ considerably from those of links in conventional wireline networks. These differences affect the behavior of network data traffic in various ways that need to be considered when designing a mobile service. As the transport protocols currently in use were originally designed for wireline networks some of them do not adapt well to wireless environments. The aspects most relevant to protocol performance are presented below. When deciding which transport protocols to use for a

7.2.5.3 Mobility

The fact that users can be on the move while using a mobile service also introduces a few aspects different from fixed environments. These differences are presented below.

A. Roaming

Roaming is the ability to move freely while maintaining an active connection to a wireless network. There are two types of roaming: contractual roaming and handover roaming.

Contractual roaming is the ability to use services outside of the network of the home service provider and still only pay the home service provider. When it comes to designing mobile services contractual roaming is not a factor.

Handover roaming is a factor that mobile services should consider. Handover roaming itself comes in two flavors, horizontal handovers and vertical handovers. Horizontal handovers are the ability to move from one access point to another within the same technology. As stated in Section 7.2.5.2 horizontal handovers have an impact for instance on network delay and on delay jitter. Vertical roaming enables moving between to different types of technologies, e.g. from UMTS to GPRS or from UMTS to WLAN. Vertical roaming might alter the behavior of a network connection completely as the characteristics of the underlying technology are changed.

B. Coverage

Another aspect of wireless networks that is not available in wireline networks is network coverage. Naturally, coverage is lost when the mobile device is moved outside of the area covered by the network, but temporary coverage loss is

also possible when moving in the network. Temporal coverage loss usually occurs in closed locations like e.g. tunnels or elevators. In addition to decreases in performance discussed Section 7.2.5.2 coverage loss often cause disconnections.

A mobile service should be able to cope with users being disconnected unexpectedly, minimizing the damage inflicted by sudden disconnections. Services preserving user state and letting users continue a previous session upon login is one example of reducing the impact of unexpected disconnections.

7.2.5.4 Other Factors

A. Switching Equipment

Due to the quite extensive range of limitations of mobile devices and mobile networks listed in the previous sections, users will prefer to use conventional devices and fixed networks whenever possible. Most users will most likely use a service from their mobile device while on the move, but then switch to the computer when at home or in the office.

A mobile instant messaging service should be able to manage the frequently changing device capabilities of users constantly changing equipment and perhaps even support simultaneous sessions from a single user with multiple devices.

B. Charging

Finally, the success of a mobile service often depends on its cost; if the service is too expensive it will not be able to compete against other similar services, e.g. mobile instant messaging cannot compete against short messaging services if it costs much more even though it comes with more advanced features.

As many operators are likely to charge by the amount of data transferred, optimizing the amount of data sent by the service not only increases the performance of the service, but also makes it cheaper to use.

7.2.6 Standards and Protocols

This section provides an overview of the standards related to mobile instant messaging. Although the xMS services cannot be classified as instant messaging, their usage is very similar to instant messaging as described in Section 7.2.6.1 and Section 7.2.6.2 introduces IMPS, currently the only instant messaging system designed specifically for mobile environments.

7.2.6.1 xMS (SMS, EMS, MMS)

Technically none of the xMS messaging services can be seen as instant messaging due to the lack of presence information. Nevertheless, the usage of these services resembles the use of instant messaging services very much.

Since users carry their mobile devices with them, their presence rate is much higher than at a fixed computer. As the probability for receiving a quick reply to a message is considerably higher, the content of xMS messages is often similar to the content in instant messages, i.e. xMS is often used for short questions to which an immediate answer is expected. The following subsections give a brief overview of the xMS messaging services.

7.2.6.2 Short Message Service (SMS)

SMS (Short Message Service) was originally designed and used for service messages, making it possible for operators to send service notifications to subscribers. This explains the low capacity (140 data octets) and limited functionality (text only) of SMS. The first SMS message was sent already in 1992, but it took until the end of the 1990's until the mass use of the SMS service begun.

At first all messages were mobile terminated, operators using SMS messages to notify users of waiting voicemail. The launch of support for mobile originated SMS messages, allowed end users to take an active role and send SMS messages to each other. This new functionality allowed the creation of interactive services using the SMS technology. Subscribers adopted this new form of mobile interaction quickly and as shown in Figure 7.4, the popularity of the SMS service has been growing swiftly ever since.

Location wise SMS is far more popular in Europe and Asia than in North America, where competing transmission technologies [159] and the Receiving Party Pays (RPP) system [160], [161] have been slowing down the rollout of SMS.

7.2.6.3 Enhanced Messaging Service (EMS)

EMS (Enhanced Messaging Service) was the first standardized messaging solution to offer richer messaging content than SMS. The EMS standard was born in the beginning of year 2000 when 3GPP, the same standardization body that defined SMS, started working on it. Of the terminal manufacturers, Alcatel, Ericsson, Siemens and Motorola have been fostering the standard and included EMS support in their terminals. The first EMS compliant mobile devices became available to customers in mid 2001.

EMS is completely based on SMS technology allowing up to 255 SMS messages to be chained together as one EMS message increasing the amount of data that can be transported in one message from 140 bytes to about 38KB. EMS supports various contents including formatted text, animation, polyphonic sounds and color pictures. EMS also allows combinations of different media types in one message as long as the maximum message size is not exceeded.

A major advantage of EMS being based on existing SMS technology is that operators do not need to invest in new infrastructure in order to offer EMS messaging. EMS messaging is completely transparent to SMS service centers. Practically the only requirement for EMS messaging is that both the sender and the recipient of an EMS message carry mobile devices that are

EMS compliant.

EMS is often seen as an intermediary step between SMS and MMS. The supporting forces behind EMS recognize the fact the MMS is a more ideal messaging service for 3G networks than EMS, however they see EMS as a cost-effective technology that eases the transition to MMS.

7.2.6.4 Multimedia Messaging Service (MMS)

Following EMS, MMS (Multimedia Messaging Service) is the next evolutionary step of mobile messaging. MMS supports messages with theoretically unlimited size and a by far richer range of multimedia content than EMS. MMS supports many standard multimedia formats like JPEG, GIF, MPEG and MIDI, which can be combined in one message using a presentation language (e.g. SMIL).

The MMS specifications are defined by both 3GPP and OMA (Open Mobile Alliance) in an effort to achieve worldwide interoperability. As opposed to EMS, which is based on SMS technology, MMS is a completely new technology. MMS makes use of existing protocols (e.g. WAP, SMTP, HTTP (Hypertext Transfer Protocol)) and message formats (SMIL, MIME) as far as possible, but requires operators to invest in new network elements. Table 7.3 summarizes the main differences between SMS, EMS and MMS.

Table 7.3: Technology wise Comparison of various mobile messaging services

Feature	SMS	EMS	MMS
Media supported	Text only	Formatted text, simple media formats, e.g. pictures, animations, sounds	Multiple rich media formats, e.g. video, audio, text
Delivery mechanism	Signaling channel	Signaling channel	Data channel
Store- and-Forward	Yes	Yes	Yes
Confirmation of message delivery	Yes	Yes	Yes
Protocols	SMS specific, e.g. SMPP	SMS specific	WAP and general Internet e.g. MIME, HTTP, SMTP
Platform	SMS Center	SMS Center	MMS Server, MMS Relay, MMS Message Store, MMS User Agent, MMS User Databases

7.2.6.5 IMPS

IMPS (Instant Messaging and Presence Service) are a set of specifications striving to ensure interoperability in mobile instant messaging. Three leading telecommunication companies, namely Ericsson, Motorola and Nokia, started the initiative in 2001. The initiative was originally known as the Wireless Village [155] but has since then been consolidated into the Open Mobile Alliance (OMA), which is an industry forum for developing market driven, interoperable Mobile service enablers.

In order to minimize interoperability issues, the initiative is open to participation from industry supporters interested in providing input regarding ongoing specification work. The aim of IMPS is not only to provide exchange of messages and presence information in mobile networks, but also to facilitate the creation of gateways to other instant messaging services.

The first approved version of the IMPS specification was released in February 2002 and it was followed up by version 1.1 in November 2002. Version 1.2 has been available as a candidate enabler since February 2003 and will be approved by the end of 2004. Currently the initiative is working on version 1.3 and a candidate enabler is planned towards the end of 2004. Although the specification work has been quite rapid, IMPS has been quite slow to market. It was not until the final quarter of 2003 that the first IMPS compliant mobile devices were released and only a few operators are currently offering instant messaging based on IMPS to subscribers. Like MMS, the IMPS service is expected to take off as soon as the penetration of IMPS capable handsets is high enough.

7.3 Entities of Presence & 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 its potential. The model for a presence service is shown in Figure 7.1a. The model for instance instant messaging is similar and is shown in Figure 7.1b. 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:

- **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 **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. A **watcher** sends requests to the presentity. 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.** It 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.** This 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.

7.4 Study of IMPS and SIMPLE Protocols

This section unit contains a comparison of two open instant messaging standards applicable in IP Multimedia Subsystem environments, namely IMPS and SIMPLE. IMPS are specifically defined for use in mobile environments and SIMPLE will soon be incorporated into 3G+ networks as part of the SIP-based IMS (IP Multimedia Subsystem). Therefore, these two services are without doubt the top contenders in the mobile instant messaging race.

As operators have not yet deployed IMPS services and IMS is not yet available in commercial wireless networks, measuring the performance of the services using tests was not possible. Hence, the comparison between the two services had to be performed on a theoretical basis. Specifications, articles and technical reports related to the area were utilized. In addition, test results from sub-areas, e.g. studies evaluating the performance of transport protocols in wireless networks, were considered in the comparison.

The aim of the comparison was to investigate which of the two standards suits the demands of mobile instant messaging the best. Therefore, the features differing between mobile and fixed environments listed in Section 7.2.5 were emphasized in the comparison. However, some features crucial to both environments, such as security, were also studied.

The comparison was performed between version 1.2 of IMPS and the current specifications of SIMPLE. IMPS version 1.2 is currently a stable candidate enabler to which no further changes are expected. The final release is planned towards the end of 2004. Most SIMPLE specifications are still ongoing work available as Internet Drafts. Nevertheless, the specifications containing the main functionality are in a stable state. Thus, although neither service is complete it is possible to carry out a fairly thorough comparison between them.

7.4.1 Functionality

The functionality is considered from the user's point of view. Succeeding sections study the technology behind the functionality in detail. This section provides an overview of IMPS and SIMPLE functionality.

Table 7.4 displays the support for main instant messaging functionality in IMPS and SIMPLE. As can be seen there are few differences between the two services. All major features are available in both services.

Table 7.4: Comparison of IMPS and SIMPLE functionality

Function	IMPS	SIMPLE
General functions		
Login and logout	Yes	Yes
Service and capability negotiation	Yes	No, services and capabilities are not negotiated with a server. It is up to client to reject unsupported requests and content types.
User search	Yes	No
Invitations	Yes	Yes
Contact lists	Yes	Yes
Presence		
Presence subscriptions	Yes	Yes
Presence notifications	Yes	Yes
Update presence	Yes	Yes
Presence authorization	Yes	Yes
Watcher list	Yes	Yes
Watcher notifications	No	Yes

Instant Messaging		
Sending and receiving instant messages	Yes	Yes
Delivery reports	Yes	Yes
Message blocking	Yes	No, must be implemented in the client
Message composition indication	No	Yes
Group Messaging	Yes	Yes, but specified by the SIPING WG

In IMPS all traffic is directed via an IMPS server before reaching its destination. The IMPS service enables clients to inform the IMPS server about supported services, client capabilities etc. This makes it possible for IMPS servers to reject traffic not supported by the client. Since SIMPLE sends instant messages peer-to-peer, a similar mechanism cannot be implemented. This explains the differences in service negotiation, capability negotiation and message blocking functionality of the two services. SIMPLE provides the means for clients to reject single messages but mechanisms for deciding which messages to reject, are not part of the specifications. SIMPLE clients can however implement e.g. message blocking in non-standard ways.

IMPS offer functionality for finding out the IMPS address of a user through a search. For example, if the real name of the user is known, the IMPS address can be discovered using the search functionality. SIMPLE does not provide this kind of functionality; the address of the user must be resolved using other means before a line of communication can be initiated.

The presence service features provided by the services are similar, except for the lack of watcher notifications in IMPS. Using watcher notifications a user can find out whenever another user subscribes to or unsubscribe from the presence information of the user. IMPS offer the functionality for retrieving a list of current watchers, but real-time notifications about changes made to the list are only sent in SIMPLE.

Message compositions indications can be used to notify a user as soon as the other party starts composing a reply to a message. This feature is available in several proprietary instant-messaging systems. SIMPLE offers support for message composition indications, while IMPS does not.

Group messaging is a feature analogous to chatting, allowing users to join groups and participate in discussions within the group. IMPS include support for group messaging. The SIMPLE working group does not directly offer group messaging. Instead, the SIPING (Session Initiation Protocol Project Investigation) working group defines a more general service called conferencing, providing group functionality for any type of media session. Therefore group messaging can be accomplished by combining conferencing with SIMPLE messaging sessions. As conferencing is included in the IMS, group messaging will be available in networks supporting the IMS architecture.

7.4.2 Evaluate Both Technologies

This section describes and compares the architectures and protocols utilized by the two compared services, IMPS and SIMPLE.

7.4.2.1 Architecture

A. IMPS

The architecture of the IMPS service is depicted in Figure 7.5 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

B. 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 7.5 lists the service elements and their main functionality. 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 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).

Table 7.5: IMPS service elements

Service element	Main functionality
Presence	Presence management Presence subscriptions Presence authorization Contact list management
Instant Messaging	Instant message delivery Access control
Group	Group usage Group management Group access control
Content	Content sharing

C. 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.

D. 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 7.6 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 4.3 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.

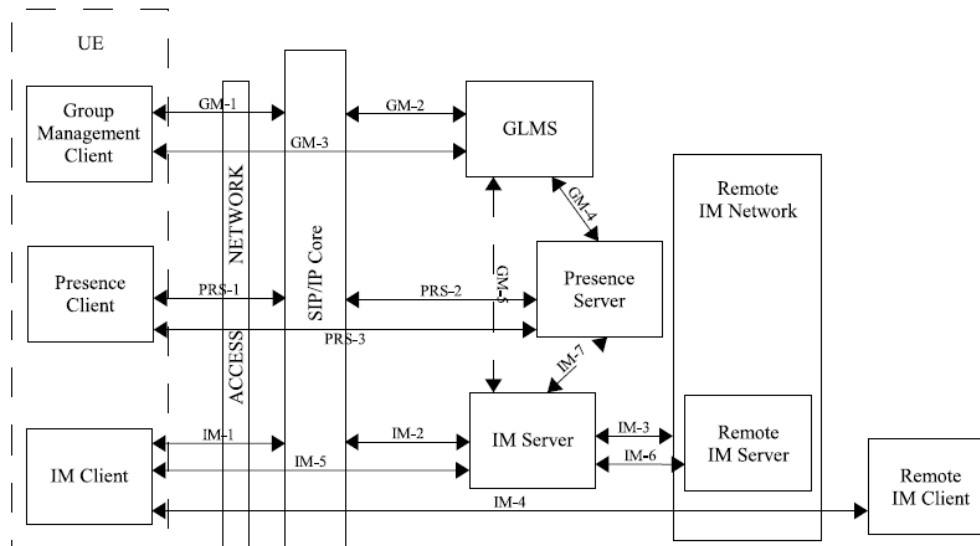


Figure 7.6: SIMPLE architecture

Table 7.6: SIMPLE reference points

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 messaging	MSRP
IM-5, IM-6	Messaging through IM Servers	MSRP
IM-7	Communication between a Presence Server and an IM Server	Undefined
PRS-1, PRS-2	Communication between a Presence Client and a Presence Server	SIP
PRS-3	Presence information and authorization	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

It should be noted that Figure 7.6 is 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 7.7). 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 7.6

E. IM Server

SIMPLE provides two modes for sending instant messages.

1. Pager mode
2. Session mode

In pager mode instant messages are sent over SIP using the MESSAGE method extension [142]. When using session mode messaging, a session is established using SIP, after which the Message Session Relay Protocol (MSRP) [162] is used for exchanging instant messages within the session.

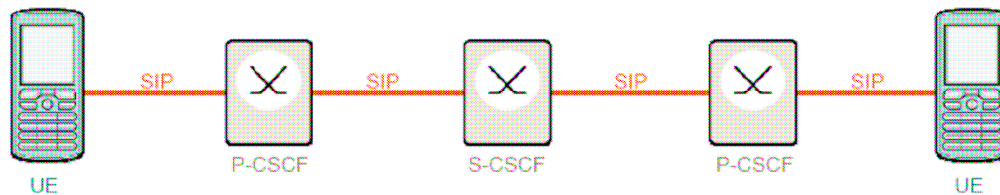


Figure 7.7 Pager mode messaging service

Both modes are able to function without any actual server functionality. In this case the IM Server element of Figure 7.6 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.

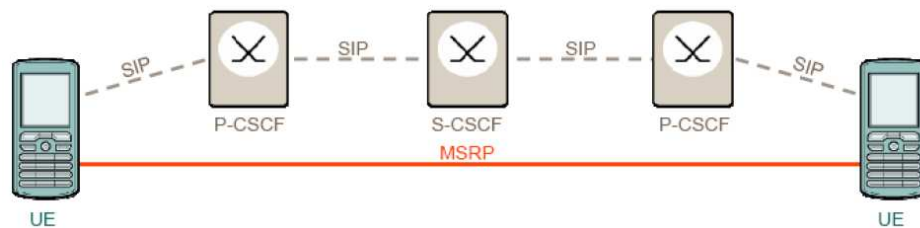


Figure 7.8 Session mode paging service

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.

F. 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 [163]. 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 7.7

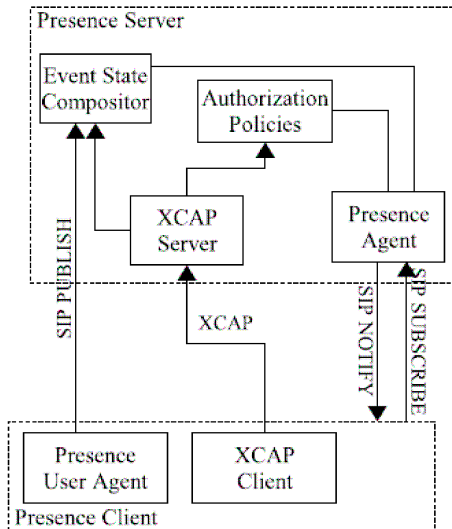


Figure 7.9: SIMPLE Presence Server

Figure 4.3 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) over PRS-3.

G. 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.

7.4.2.2 Protocols

This section presents protocols specific to the IMPS and SIMPLE services. Well-known protocols such as TCP, UDP or HTTP are not presented although used by the services. Only protocols enabling the actual functionality of the services are presented.

A. IMPS

The IMPS architecture and the protocols used for transporting data between the elements of the architecture are presented in Figure 7.6. This section presents CSP and SSP, the protocols most essential to the IMPS service of the other protocols; CLP is quite inconvenient to operate and contains only a subset of the functionality of CSP. The Server Mobile Core Network Protocol (SMCNP) for enabling features such as charging has not been defined by OMA at all.

B. Client-Server Protocol (CSP)

The Client-Server Protocol (CSP) provides IMPS clients with access to the IMPS server and its functionality. The functions provided by the server are used through CSP transactions. A CSP transaction consists of the messages needed to carry out a function, ordinarily one request and its response. Most CSP transactions are only available within a CSP session. A CSP session is established when the user logs in to the system and terminated upon user logout or if the server decides to disconnect the user. CSP sessions are transport independent, i.e. they remain valid even when the transport connection is broken, a phenomenon typical to mobile networks as described in Section 7.2.5.3.

In order to achieve the flexibility needed for CSP to be used in clients varying from mobile terminals to fixed PC-clients, several different transport bindings have been defined. Logically a CSP transport binding is divided into two channels: a mandatory data channel in which all CSP messages are exchanged and a conditional CIR (Connection Initiation Request) channel used to activate the data channel whenever it is not active (Figure 7.8).

This communication model enables a server to communicate with clients behind proxies, which is a usual situation in mobile networks as per Section 7.2.5.4

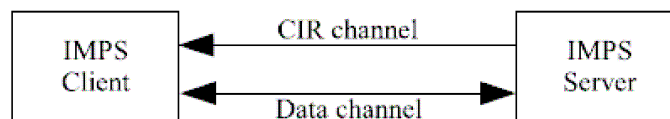


Figure 7.10: IMPS communication model

The protocol bindings defined for the data channel are WSP (Wireless Session Protocol), HTTP, HTTPS and SMS, while the bindings for the CIR channel are WAP push over SMS, WAP push over UDP, SMS, UDP, TCP and HTTP.

CSP data is carried over the network using the XML message format according to the DTDs (Document Type Definition) specified in [164] and [165]. In order to optimize messages for size, CSP also supports the WBXML (Wireless Binary XML) format [166].

C. Server-Server Protocol (SSP)

The Server-Server Protocol (SSP) [167] connects IMPS servers with each other. SSP enables IMPS clients to use IMPS functionality distributed across the network, possibly provided by different service providers. In addition, SSP enables other instant messaging networks to communicate with the IMPS network through proprietary gateways.

SSP is transferred using either HTTP or HTTPS over the TCP transport protocol. As HTTP is an asymmetrical protocol two physical TCP connections are required in order to provide two-way communication. SSP uses persistent TCP connections for improved performance. Equally to the CSP protocol, SSP is also conveyed between network elements in the XML format. WBXML is not supported by SSP since servers ordinarily communicate with each other via wireline connections with high bandwidth.

D. SIMPLE

Session Initiation Protocol (SIP)

The Session Initiating Protocol (SIP) is a signaling protocol used for creating, modifying and terminating sessions with one or more participants []. SIP allows for any kind of session to be initiated, including IP telephony calls, multimedia conferences and instant messaging sessions. Sessions created using SIP are peer-to-peer, i.e. the SIP framework is only used for managing sessions, not for session traffic. This typically results in a SIP trapezoid shown in Figure 7.11.

User agents are SIP entities able send SIP requests acting as clients (UAC) and respond to incoming requests as servers (UAS). All servers and clients of the SIMPLE architecture (Figure 7.11) act as user agents.

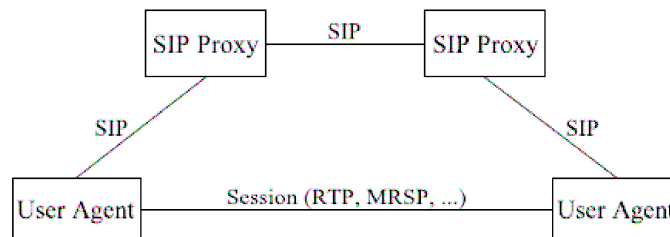


Figure 7.11: The SIP trapezoid

SIP proxies accept SIP requests from user agents or other SIP proxies and route them closer to the recipient.

In addition to proxies and user agents, registrars are also important elements in SIP networks. Registrars accept registrations of address information from other SIP entities. The registrars store the information in location services, which are then used by SIP proxies when routing messages.

Finally, redirect servers can be utilized to redirect incoming requests to other destinations, e.g. a user could set the SIP phone at work to redirect all calls to the SIP phone at home.

SIP is very similar to the HTTP protocol [22], utilizing the same request/response transaction model where each transaction consists of a request invoking a particular method and the responses to the request. The majority of the message and header field syntax is also derived directly from HTTP.

SIP allows extensions to the protocol to be made in the form of new methods and header fields. Three extensions of the SIP protocol have a central role in SIMPLE, namely the event notification extension, the pager mode instant messaging extension [14] and the UPDATE method extension. The event notification extension enables SIP nodes to be notified when a particular event occurs.

The instant messaging extension adds support for sending instant messages in SIP messages without creating a session. Finally, the UPDATE method extension provides the means for updating presence information. In addition, group messaging software may use the REFER method extension.

Table 7.7 lists the main SIP methods and their use in SIMPLE.

E. Message Session Relay Protocol (MSRP)

The Message Session Relay Protocol (MSRP) is an instant message transport protocol defined by the SIMPLE working group. As opposed to pager mode instant message exchange, which uses the SIP MESSAGE method, MSRP is a session-based protocol. MSRP sessions can be setup using any signaling protocol capable of initiating sessions. In SIMPLE the SIP protocol is naturally used.

Like SIP, MSRP is also a text-based protocol similar to the HTTP protocol. MSRP is a fairly simple protocol, only two request methods are needed: the SEND method and the REPORT method. The SEND method is obviously used for sending instant messages. MSRP does not limit instant messages to contain only text; any content type can be used. The REPORT method can be used for receiving delivery reports for sent instant messages.

Table 7.7: SIMPLE usage of SIP methods

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

As MSRP is based on HTTP, it allows for extending the protocol by adding new methods and header fields. The 'Relay Extensions for MSRP' specification does exactly that, adding support for using relay intermediaries to MSRP, which originally is a peer-to-peer protocol.

F. XML Configuration Access Protocol (XCAP)

The XML Configuration Access Protocol (XCAP) [170] allows clients to retrieve, modify and delete XML data stored on a server. Although defined by the SIMPLE working group, XCAP is a general-purpose protocol which area of usage is not restricted to SIMPLE. However, XCAP is particularly suitable for the needs of SIMPLE as all data permanently stored in SIMPLE server elements, such as presence information, watcher information and presence authority rules, is defined in the XML format.

XCAP provides the means for mapping XML documents and document elements directly to HTTP URIs (Uniform Resource Identifier). This enables XCAP to manipulate XML documents stored on a server using normal HTTP primitives. The HTTP GET method is used for retrieving XML documents or parts of them, the PUT method is used for creating or modifying documents and the DELETE method can be utilized for deletion of XML documents.

The XCAP specification in [170] defines a framework for manipulation of XML data. In addition, each application storing data on XCAP servers comes with application-specific conventions, e.g. XML schemas and bootstrap URIs, which need to be defined. Therefore each application using XCAP is required to specify application-specific requirements in an 'XCAP Application Usage' document. For example, an application usage for manipulating presence information using XCAP is defined in [171].

7.4.3 Culmination for further implementation of Research

The architecture of IMPS is more mature than SIMPLE architecture. The IMPS architecture is relatively simple; complete instant messaging functionality can be delivered using only two protocols, CSP & SSP, defining the client-client and the server-server interfaces clearly & unambiguously.

The SIMPLE architecture is without doubt more complex; numerous reference points are defined for accessing the service. The functionality has been divided into several small elements of which each has been defined independently from the others in order to provide a flexible system. However, although the client-server interfaces of the elements are quite clearly defined, communication between server elements is mostly undefined. Also, on several occasions the SIMPLE service provides several alternatives for performing the same function. While this on one hand improves flexibility, it might on the other hand increase complexity as server solutions are forced to provide all alternatives in order to be functional against all kinds of clients. Again, it should be stressed that the SIMPLE specifications are ongoing work; changes to the current solution are possible.

The protocols on which the network communication of IMPS and SIMPLE relies are very similar; both services rely on HTTP-like protocols based on the request/response model. In addition, all data except for the content of instant messages is transferred in the XML format for both IMPS and SIMPLE. The characteristics of the protocol and data formats when it comes to performance, security etc. are evaluated in subsequent sections.

7.5 The Common Profile for Instant Messaging (IM)

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: 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 7.12.

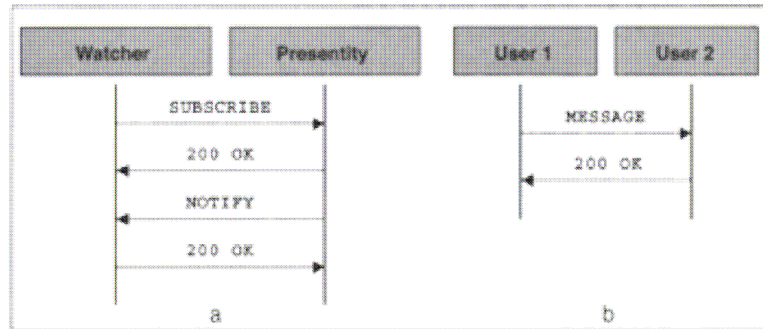


Figure 7.12 CPIM models for presence and instant message services.

Figure 7.12a shows the message exchange to subscribe and Figure 7.12b shows the basic message exchange for a subscribed user. There also is a corresponding Unsubscribe message, not shown in Figure 7.12.

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 Internet work directly, depending on their implementation.

7.6 Example of 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:friend1@ispl.com'
    target='pres:friend2@isp2.com'
    duration='64000' transID='I' />
  
```

Response

The response is:

```

<response status='success' transID='I' 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:friend1@ispl.com'
      target='pres:friend2@isp2.com'
      transID='I' />
<presence entityInfo='http://www.isp2.com/friend2' />
<tuple destination='im:friend2@isp2.com'
      status='open' />
</notify>
```

Unsubscribe

The unsubscribe is:

```
<unsubscribe watcher='pres:friend1@ispl.com'
      target='pres:friend2@isp2.com'
      transID='I' />
and, if successful, will be informed:
<reponse status='success' transID='I' />
```

7.7 Example of 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='I' />
```

Content-Type: text/plain; charset="us-ascii"

Hello! How are you?

Response

The response is:

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

In case of success, the response will be:

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

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

7.8 Performance evaluation of IMPS and SIMPLE

This section evaluates the performance of IMPS and SIMPLE compared to each other. Several aspects such as bandwidth usage, delays, processing, coverage loss, proxy traversal and scalability were studied. As mentioned earlier, the performance comparison is based purely on theoretical facts; no real tests could be performed due to the lack of deployed implementations of the services.

7.8.1 Bandwidth Usage

As instant messages usually only contain short strings of text, it might seem unimportant to optimize them for size. However, in addition to the actual data, instant messages carry addressing information, session identifiers etc. increasing the size of the message substantially. Moreover, although IMS networks offer a broad bandwidth, only a fraction of it will be allocated for SIP signaling. Also, it will take long until complete IMS network coverage is reached and until then low-bandwidth technology, such as GPRS, will be widely used.

A. Message Size

Protocol messages consist of two parts: protocol data and payload data. In order to optimize the total size of a message, both parts need to be optimized. IMPS messages have a relatively small protocol part; most information is generally carried as payload data. SIMPLE messages carry more information in the protocol part of the message and only the actual instant message or presence information is carried in the payload part.

For optimizing the size of the protocol part IMPS includes a WSP (Wireless Session Protocol) transport binding. In essence, WSP is a tokenized version of HTTP resulting in smaller sized messages. For compressing the payload data, IMPS provides support for the WBXML format. WBXML tokenizes the XML elements of the CSP protocol. As the XML elements of the CSP protocol have lengthy names, the WBXML format reduces the size of the payload data significantly. On average the size of a compressed CSP message is about 25% of the uncompressed message.

SIMPLE SIP messages can be compressed using the SigComp solution [172]. SigComp compresses both the protocol and the payload part of SIP messages. A performance evaluation performed by Nordberg et al. in [173] indicates that the message size can be reduced to approximately 25-50% of the uncompressed size for SIP messages sent in 3G networks.

By comparing the message sizes of instant messages for both services (see Appendix A for message examples) it can be noted that sending uncompressed IMPS instant messages consumes over three times more data

than pager mode SIP instant messages. Setting up a SIMPLE session mode and sending one message requires a little more data than one IMPS message. Nevertheless, after session mode has been set up, all subsequent SIMPLE instant messages are over five times smaller than corresponding IMPS messages. Conclusively, despite IMPS providing a somewhat higher compression rate, SIMPLE messages are still considerably smaller than IMPS messages.

Finally, the message size can also be affected by reducing the size of presence notifications received from the presence server. The size of notifications can be decreased using partial notifications, i.e. the server sends only the changed part of the presence information in the notification. Both IMPS and SIMPLE includes support partial notifications.

B. Filtering Rules

The bandwidth usage can also be limited using filtering rules. Clients can use filtering rules to narrow the amount of events resulting in notifications. Filtering rules can also be used to block instant messages from certain users, thus limiting the amount of data received. IMPS support filtering rules for both presence information and instant messaging. SIMPLE only supports filtering rules for presence information; instant messages cannot be blocked before arriving at the client due to the peer-to-peer property of SIMPLE.

C. Signaling

IMPS needs no signaling in order to send an instant message to another user. SIMPLE needs to setup a session using SIP when using session mode instant messaging, thus introducing data overhead in comparison to IMPS.

Furthermore, SIMPLE presence subscriptions are not permanent clients must refresh subscriptions periodically. The length of the period depends on the policy of the presence server. In comparison, IMPS presence subscriptions are guaranteed to last for the length of an IMPS session. However, IMPS clients might need to renew subscriptions whenever they login to the system, creating overhead comparable to SIMPLE presence refreshments.

7.8.2 Delays

Delays and delay jitter are not as important factors in instant messaging as in other real-time services. E.g. when sending voice or video, keeping delays and delay jitter to a minimum is critical to the quality of the communication. Nevertheless, instant messaging is a real-time service and if delays grow too high, the user experience deteriorates. Delay jitter is not a factor in instant messaging since messages are not sent in an uninterrupted stream.

There are two types of delays affecting user experience: the session setup delay and the message exchange delay. These are studied in more detail below.

A. Session Setup

The session setup delay is the amount of time needed for initiating the sending of an instant message. The delay caused from establishing a data channel to the radio access network can be neglected here as both services perform the same procedure.

In IMPS, clients create a session with the IMPS server upon logging in to the system, but sessions are not initiated with recipients of instant messages. Hence there is no session setup delay associated with the IMPS service.

SIMPLE pager mode messaging generates no session setup delay either. SIMPLE session mode messaging requires a session to be set up using SIP before the first instant message can be sent. A minimum of three SIP messages or 2 round-trip times (RTTs) is needed to complete the setup of a SIMPLE instant messaging session. Functionality for restricting the size of a single message in order to prevent clients from running out of memory.

IMPS clients can specify the maximum message size it is able to process in one chunk during the login procedure. In SIMPLE the MSRP protocol allows sending large messages in smaller chunks when using session based messaging. For pager mode, instant messages are restricted to 1300 bytes by the specification.

7.8.3 Coverage Loss

Both the IMPS and the SIMPLE service are able to cope with sudden disconnections caused by the loss of radio network coverage. Clients of neither service need to re-login to the system upon a disconnection, only the transport layer connections need to be reestablished. For session based messaging, SIMPLE clients need to store some information in order to be able to re-initiate the transport connection upon sudden disconnections, otherwise a new session must be negotiated.

7.8.4 Proxy Traversal

Mobile networks are usually connected to the Internet via NAT proxies. Therefore, the ability to communicate through proxies is an important feature of mobile instant messaging services as it enables a global service, available from both mobile and wireline clients.

An IMP makes use of the CIR channel in order to traverse proxies. As all client-server communication is client-initiated and the server can trigger client requests using the CIR channel, IMPS clients can use the IMPS service through proxies without problems.

Implementations of SIP as specified in [168] have severe problems with NAT traversal. The problem stems from SIP proxies sending replies to ports defined in the SIP requests. RFC 3581 defines a header extension to the SIP protocol solving the problem for TCP. SIMPLE services function perfectly inside mobile networks, e.g. using IMS in 3G networks, but due to the NAT problems it is currently not possible to create fully functional SIMPLE services spanning both mobile and wireline networks.

7.8.5 Scalability

By scalability, the system performance when used by a substantial amount of simultaneous users is meant. Scalability is important as these services are expected to acquire millions of users over time. The architectures of the two compared services provide excellent scalability. Neither service requires any centralized network elements, enabling data to be distributed among several servers. By building a network of distributed servers, both services can serve a vast amount of simultaneous clients.

Generally, SIMPLE performs better than IMPS when it comes to optimizing bandwidth usage and delays. IMPS cause no session setup delays, but session mode messaging enables shorter message exchange delays and smaller messages for SIMPLE. On the other hand, the mechanisms causing delays in message exchange for IMPS also enable IMPS messages to traverse NAT proxies without problems. In contrast to IMPS, SIMPLE currently has severe problems with NAT traversal.

The architectures of both services enable good scalability supporting customer bases of large scale. The ability to recover from disconnections due to coverage loss is also good for both IMPS and SIMPLE.

7.9 Implementation

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.

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.

7.9.1 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.

7.9.2 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 compliant for as many platforms as possible according to the requirements.

7.10 Results

Output results are attached here in the form of testing logs

7.10.1 Logs for login logout

2009-02-21

11:45:19:601,User=mobile,Action=Login,IP=172.21.111.124,SessionID=49E666F6.00000000.mobile,SessionType=IMPS

2009-02-21

11:45:45:101,User=mobile,Action=Logout,IP=172.21.111.124,SessionID=49E666F6.00000000.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 night: 128 Content

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 Connection

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 4E4F 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.imstestbed.net/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.imstestbed.net /CSP1.1">
<Session>
<SessionDescriptor>
<SessionType>

```

Outband

```

</SessionType>
</SessionDescriptor>
<Transaction>
<TransactionDescriptor>
<TransactionMode>

```

Response

```

</TransactionMode>
<TransactionID>
nok1
</TransactionID>
</TransactionDescriptor>
<TransactionContent xmlns="http://www.imstestbed.net /TRC1.1">
<Login-Response>
<ClientID>
<URL>
WV:IMPEC01$00001@NOK.S60
</URL>
</ClientID>
<Result>
<Code>
200</Code>

```

```

<Description>
Login OK</Description>
</Result>
<SessionID>49E666F6.000000000.mobile
</SessionID><KeepAliveTime>
90</KeepAliveTime>
<CapabilityRequest>
T</CapabilityRequest>
</Login-Response>
</TransactionContent>
</Transaction>
</Session>
</WV-CSP-Message>
2009-02-21      11:45:18,mobile,172.21.111.124,nok1,ToServer,Login-
    Request,mobile
2009-02-21      11:45:18,mobile,172.21.111.124,nok1,ToClient,Login-
    Response,401
2009-02-21      11:45:19,mobile,172.21.111.124,nok1,ToServer,Login-
    Request,mobile
2009-02-21      11:45:19,mobile,172.21.111.124,nok1,ToClient,Login-
    Response,200
2009-02-21 11:45:21,mobile,172.21.111.124,nok2,ToServer,ClientCapability-
    Request
2009-02-21 11:45:21,mobile,172.21.111.124,nok2,ToClient,ClientCapability-
    Response
2009-02-21 11:45:22,mobile,172.21.111.124,nok3,ToServer,Service-Request
2009-02-21      11:45:22,mobile,172.21.111.124,nok3,ToClient,Service-
    Response
2009-02-21 11:45:23,mobile,172.21.111.124,nok4,ToServer,GetList-Request
2009-02-21 11:45:23,mobile,172.21.111.124,nok4,ToClient,GetList-Response
2009-02-
    2111:45:25,mobile,172.21.111.124,nok5,ToServer,UnsubscribePresence-
    Request,wv: mobile/~IM1.0_friendslist
2009-02-21 11:45:25,mobile,172.21.111.124,nok5,ToClient,Status,200
2009-02-2111:45:26,mobile,172.21.111.124,nok6,ToServer,ListManage-
    Request,wv:mobile/~IM1.0_friendslist
2009-02-21      11:45:26,mobile,172.21.111.124,nok6,ToClient,ListManage-
    Response,200

```

2009-02-21 11:45:28, mobile, 172.21.111.124, nok7, ToServer, CreateAttributeList-Request, DefaultList
 2009-02-21 11:45:28, mobile, 172.21.111.124, nok7, ToClient, Status, 200
 2009-02-21 11:45:29, mobile, 172.21.111.124, nok8, ToServer, UpdatePresence-Request, "OnlineStatus, ClientInfo, CommCap, UserAvailability, StatusText, StatusContent"
 2009-02-21 11:45:29, mobile, 172.21.111.124, nok8, ToClient, Status, 200
 2009-02-21 11:45:30, mobile, 172.21.111.124, nok9, ToServer, Polling-Request
 2009-02-21 11:45:31, mobile, 172.21.111.124, nok9, ToServer, GetBlockedList-Request
 2009-02-21 11:45:31, mobile, 172.21.111.124, nok9, ToClient, GetBlockedList-Response
 2009-02-21 11:45:33, mobile, 172.21.111.124, nok10, ToServer, Search-Request, GROUP_USER_ID_OWNER=mobile
 2009-02-21 11:45:33, mobile, 172.21.111.124, nok10, ToClient, Search-Response, SearchFindings=0
 2009-02-21 11:45:35, mobile, 172.21.111.124, nok11, ToServer, SubscribePresence-Request, wv:mobile/~IM1.0_friendslist
 2009-02-21 11:45:35, mobile, 172.21.111.124, nok11, ToClient, Status, 200
 2009-02-21 11:45:35, mobile, 172.21.111.124, wvnow4419a6b5, ToClient, PresenceNotification-Request
 2009-02-21 11:45:36, mobile, 172.21.111.124, nok12, ToServer, Polling-Request
 2009-02-21 11:45:36, mobile, 172.21.111.124, nok12, ToServer, StopSearch-Request
 2009-02-21 11:45:36, mobile, 172.21.111.124, nok12, ToClient, Status, 200
 2009-02-21 11:45:36, mobile, 172.21.111.124, nok13, ToServer, Polling-Request
 2009-02-21 11:45:37, mobile, 172.21.111.124, wvnow4419a6b5, ToServer, Status, 200
 2009-02-21 11:45:38, mobile, 172.21.111.124, nok14, ToServer, Polling-Request
 2009-02-21 11:45:43, mobile, 172.21.111.124, nok13, ToServer, UpdatePresence-Request, CommCap
 2009-02-21 11:45:43, mobile, 172.21.111.124, nok13, ToClient, Status, 200
 2009-02-21 11:45:45, mobile, 172.21.111.124, nok14, ToServer, Logout-Request
 2009-02-21 11:45:45, mobile, 172.21.111.124, nok14, ToClient, Disconnect, 200

7.11 Comparison of IMPS & SIMPLE Performance

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 [5].

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 are 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 7.8 and 7.9 list the main advantages and disadvantages of the compared services.

Table 7.8: Advantages and disadvantages of IMPS

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

Table 7.9: Advantages and disadvantages of SIMPLE

Advantages	Disadvantages
Scalability	Complex architecture
Interoperability	Proxy reversal
Security	
Efficiency	

7.11.1 Some User scenarios of Presence Service

Several user scenarios had been envisioned through the work of the Open Mobile Alliance (OMA), which is the body in charge to standardise the services and applications built on the top of the architecture and technology defined by the 3GPP.

Most of the requirements involve the use of Presence Service as an improvement of Instant Messaging and the user contact list (the phone address book), which allows seeing the willingness and availability of the contact to establish a communication. Moreover, the presence information indicates the media, which the contact wants to use for the communication (Instant Messaging, SMS, email, fixed phone, mobile, etc.)

However, some user scenarios require that the users configure their terminals manually to expose adequate presence information. Using a set of profiles does this. For example, if the user arrives at her office, she would have to change her profile on her mobile device to Office. Profile, which means that people who want to contact her will see her presence information as .I want to be contacted by my office phone and only emergency calls can go through my mobile.

Other user scenarios such as advertisements and the subscription of alerts (events, news, weather, traffic, etc.) require the update of presence information through different networks as well as require that the user set complex rules and preferences for blocking users and filtering presence information. Thus ways to avoid this must also be investigated.

In the following scenario: a user has her mobile device and is entering to a large mall, which is providing access to its wireless network. As soon the user is entering the mall, her presence information needs to be updated. This presence information can include additional information like the mall card number of a frequent buyer. Because the mall is able to track the customer habits of shopping using this number, the user can receive alerts of interesting offers for her into her mobile device.

This user could need help to locate a shop or particular product; or she may be lost in the mall; therefore she would require contacting the customer help desk. Logically this kind of communication or access to a service provided by the mall must be free of charge for the users; therefore the user will see in her phone book a new temporal contact with presence information such as public mall help with busy/available then the user can use her PoC mobile application and start to communicate with the mall staff. Moreover, this communication can include other multimedia content such as a map or video. This is an example of the future evolution of PoC, which is called Push to X.

Presence information can improve existing corporate directories, which can also be used to advertise services. A powerful search engine can exploit these directories to make it easy to find people and services. This will be more obvious in the future when users may broadly use presence-aware applications.

In the world of collaborative work such as collaborative writing or brainstorm meetings (face-to-face, using videoconference or Instant Messaging). Several user scenarios can be enhanced with document presence information. For instance, in a meeting face-to-face all the participants may carry a PDA and form an ad-hoc network like Bluetooth, wi-fi etc., which allow them to share documents along with document presence information.