INVESTIGATING THE VIABILITY OF A FRAMEWORK FOR SMALL SCALE, EASILY DEPLOYABLE AND EXTENSIBLE HOTSPOT MANAGEMENT SYSTEMS

Mamello P. Thinyane

DEPARTMENT OF COMPUTER SCIENCE

Rhodes University, Grahamstown

This work is submitted in fulfilment of the requirements for the degree of Master of Science in Computer Science at Rhodes University.

Abstract

The proliferation of PALs (Public Access Locations) is fuelling the development of new standards, protocols, services, and applications for WLANs (Wireless Local Area Networks). PALs are set up at public locations to meet continually changing, multi-service, multi-protocol user requirements. This research investigates the essential infrastructural requirements that will enable further proliferation of PALs, and consequently facilitate ubiquitous computing. Based on these requirements, an extensible architectural framework for PAL management systems that inherently facilitates the provisioning of multiple services and multiple protocols on PALs is derived. The ensuing framework, which is called Xobogel, is based on the microkernel architectural pattern, and the IPDR (Internet Protocol Data Record) specification. Xobogel takes into consideration and supports the implementation of diverse business models for PALs, in respect of distinct environmental factors. It also facilitates next-generation network service usage accounting through a simple, flexible, and extensible XML based usage record. The framework is subsequently validated for service element extensibility and simplicity through the design, implementation, and experimental deployment of SEHS (Small Extensible Hotspot System), a system based on the framework. The robustness and scalability of the framework is observed to be sufficient for SMME deployment, withstanding the stress testing experiments performed on SEHS. The range of service element and charging modules implemented confirm an acceptable level of flexibility and extensibility within the framework.

Acknowledgements

I would like to acknowledge and thank my supervisors, Dr Greg Foster and Professor Peter Clayton for the guidance and direction they offered in this project.

I thank my family for their immesurable love, for being the happiest family I know and for the constancy of their support in my life (and in this project ☺).

My friends deserve my gratitude for walking with me each day.

The financial assistance from the Andrew Mellon scholarship towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to Rhodes University or the donor.

This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom, Business Connexion, Converse, Verso Technologies, and THRIP.

My absolute gratitude goes to the sovereign God for His mercy [Romans 9:15].

Table of contents

List of Figures

Table of listings

List of tables

1 Chapter 1: Introduction

Public Access Locations (PALs), defined as public locations where wireless network access is available, are growing in eminence to provide ubiquitous connectivity to users. PALs are currently implemented at airports, coffee shops, educational institutions, recreational areas and shopping malls. The PALs market is slowly emerging out of the build-up/growth phase into the mature phase with the worldwide number of PALs expected to grow from 43 850 in 2003 to 200 000 in 2008 (ITFacts-1, 2004). Gartner[1] reported that the number of worldwide PAL users increased from 9.3 million in 2003 to 30 million in 2004 (ITFacts-2, 2005) and it is predicted to be 120 million by 2008. The proliferation of PALs has been stimulated by the need to access data on the move and by the trend to always be connected. In addition, the proliferation of PALs has also been facilitated by the maturing and the advancements in the wireless networking technologies and standards. Currently there exists a plethora of different wireless networking technologies each catering for different environmental conditions and implementations.

1.1 Wireless networking technologies

Wireless communication technology can be traced back to 1894, when Guglielmo Marconi began experimenting with radio waves in an attempt to produce and to detect radio waves over a long distance (JHSPH, 1999). The technology has since been adopted by governments, businesses, schools and other institutions to add value to their communication infrastructure. Wireless networks differ from the wired counterparts in the way they handle the lower layers of the Open Standard Interconnect (OSI) model (i.e., the transmission of the data between communicating devices). The current methods of wireless signal transmission include infrared, laser, narrowband microwave and spread-spectrum (JHSPH, 1999). There are different wireless networking protocols depending on the differences in their implementation (e.g., transmission mechanism, signal range, signal strength) and these differences determine the applicability of these protocols in PAL service provisioning. Some of

[1] Gartner is a leading provider of research and analysis on the global IT industry. Online at http://www.gartner.com/

the current wireless protocols are outlined below, highlighting the main features and the specific circumstances in which they are implemented.

1.1.1 Mobile (Cellular) technologies

These are the wireless technologies used extensively for mobile and cellular telephony. They represent different generations of wireless technologies (e.g., GSM is a second generation (2G) technology and EDGE is a third generation (3G) technology).

1.1.1.1 GSM

General System for Mobile communication (GSM) is a 2G digital radio standard developed by the European Telecommunications Standard Institute (ETSI) which runs at 9.6 kbps. It is a standard that was developed to operate in the high user density with low-power mobile stations. GSM is used extensively for voice and data transfer of up to 9.6 kbps. GSM operation is very similar to normal radio communication in that signals are sent using free standing transmitters, called base stations. The Physical (PHY) layer in GSM is provided by radio transmission (Halonen et al, 2003).

1.1.1.2 GPRS

General Packet Radio Service (GPRS) is another standard for wireless communication that runs at 115 kbps. It is a non-voice service that provides transmission of Internet Protocol (IP) data packets over existing cellular networks. Any service available on the internet (e.g., e-mail, web browsing, Internet Relay Chat (IRC)) is available to GPRS-enabled devices. GPRS is implemented as a 2.5G technology and offers enhancements, in terms of data transmission and efficiency, to the 2G technology (Halonen et al, 2003) GPRS uses radio transmission for its PHY layer and a suite of other protocols at the data link layer to ensure reliable transfer of message between network nodes.

1.1.1.3 EDGE

Enhanced Data rates for Global Evolution (EDGE) is a 3G technology that provides up to 384 kbps data access to mobile devices. EDGE can be viewed as a packet-switched enhancement to GPRS, called Enhanced GPRS (EGPRS), and as a circuit-switched data enhancement called Enhanced Circuit-Switched Data (ECSD). EDGE

introduces a new modulation technique and channel coding mechanism that allows transmission of packet-switched and circuit-switched voice and data. EDGE runs the same packet handling protocols as GPRS at the core network and only introduces differences at the base station system side due to the new transceiver unit that handles EDGE modulation (Halonen et al, 2003).

1.1.1.4 UMTS

Universal Mobile Telecommunications Services (UMTS) is a 3G broadband wireless standard that offers packet-based transmission of text, voice, and multimedia data. UMTS runs at speeds of up to 2 Mbps. UMTS provides an increase in network capacity, data speeds and service capabilities relative to 2G mobile networks. The three domains that make up a UMTS network are: Core network which is responsible for switching, routing and storage of databases and network management functions, UMTS Terrestrial Radio Access Network (UTRAN) which provides the interface access method to the user equipment, and the User Equipment which is the mobile device that access the UMTS network (Patil et al, 2003).

1.1.2 IEEE 802.x standards

The Institute of Electrical and Electronic Engineers (IEEE) has established a number of wireless communication protocols and standards which are defined within the 802 suite of protocols. The prominent of these protocols are:

1.1.2.1 IEEE 802.15

IEEE 802.15 which is also known as Bluetooth, is a Wireless Personal Area Network (WPAN) radio standard that addresses short distance (~10 meters) networking requirements for portable, low power mobile computing devices. It is currently used extensively for connecting cell phones, PDAs, audio headsets, computer keyboards etc. Bluetooth operates in the unlicensed (Industrial, Scientific and Medical) ISM bands at 2.45 GHz (Naeve M., 2005) and provides data rates of up to 751 kpbs**.** The two modules that provide the data transmission functionality of the lower OSI layers are the Radio module and the Link Manager module. Some of the other protocols in the Bluetooth protocol stack are the Logical Link Control and Adaptation Protocol (L2CAP), the Host Control Interface (HCI) and the Service Discovery Protocol

(SDP). The higher level protocols and application layer protocols are then responsible for the software-related data handling functionality.

1.1.2.2 IEEE 802.15.3a

IEEE 802.15.3a, which also goes by the name WiMedia or ultrawideband (UWB), is a high-speed WPAN standard for low powered devices. The Federal Communications Commission (FCC) has assigned the 3.1 GHz to 10.6 GHz spectrum as the operating frequency range for UWB. The fastest data rate observed on UWB is 252 Mbps and the theoretical max data speed is 480 Mbps. The UWB, along with a convergence layer, is set to provide an underlying common radio transport mechanisms for many different other applications, including Wireless Universal Serial Bus (WUSB), IEEE 1394, next generation Bluetooth and Universal Plug and Play (UPnP) (Kolic R., 2004).

1.1.2.3 IEEE 802.15.4

IEEE 802.15.4 also known as Zigbee, is a low data rate (20 – 250 kbps) specification that is specifically designed for low power devices and that operates in the unlicensed ISM radio bands. The standard has been developed for remote monitoring and controlling of devices. It defines transmission and reception on the PHY layer, Personal Area Network (PAN) maintenance, and reliable data transport Medium Access Control (MAC) layer. Zigbee is designed for mesh networking and it allows for more nodes and a greater range than the 802.15 specification (Frank R., 2004).

1.1.2.4 IEEE 802.11

IEEE 802.11, known as Wireless Fidelity (WiFi), is a group of Wireless Local Area Network (WLAN) standards that use Ethernet protocol, hence why the standards is sometimes called wireless Ethernet. WiFi was developed to "provide wireless connectivity to automatic machinery, equipment, or stations that require rapid deployment, which may be portable or hand-held, or which may be mounted on moving vehicles within a local area” (Hayes V., 1990).

The original 802.11 standard specified three transmission methods for the PHY layer; Infrared, Frequency Hoping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS) (O'Hara et al, 1999). The infrared transmission uses light spectrum, and FHSS and DSSS use the unlicensed 2.4 GHz and 5 GHz radio frequencies (RF) in the ISM bands. FHSS involves transmitting data in a range of frequencies using a specified hop sequence. DSSS on the other hand involves sending the data simultaneously over several frequencies. The standard also specifies MAC sub-layer protocols implemented at the data link layer. The current revisions to the standard are 802.11a, 802.11b, and 802.11g.

1.1.2.4.1 802.11a

The standard came out in 1999 and was designed to run at 54Mbps. In order to operate at that speed, the delay spread problem[2] had to be countered and this was done by using the Orthogonal Frequency Division Multiplexing (OFDM) modulation technique and operating at the 5GHz RF spectrum. 802.11a has not attained a higher uptake due to of the following reasons:

- The 802.11b specification had already gained popularity and more devices were already shipped with the 802.11b chips. The 802.11a specification is not interoperable with 802.11b.
- There was a slow availability of 5GHz components needed for the development of the chips.
- The range of the 802.11a specification is shorter than the 802.11b specification (Reynolds J., 2003).

1.1.2.4.2 802.11b

This revision was released in 1999 and operates in the 2.4GHz RF spectrum at raw speeds of up to 11Mbps. 802.11b uses the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) method defined in the original 802.11 specification. This revision gained acceptance as the WLAN standard due to more 802.11b chipsets coming into the market (the DSSS modulation technique is used in this specification and this made it easy for chip manufacturers to upgrade from the chips made for the

[2] This is the problem that occurs when a signal bounces off objects and echoes off walls. It results in the same signal reaching the receiver at different times.

original specification). The range of the protocol can be extended to approximately 8 km with fixed wireless point-to-point architecture and a usage of high gain antennae.

1.1.2.4.3 802.11g

This standard got released in June 2003 and it operates at the same frequency as the 802.11b specification (i.e., 2.4 GHz) with the raw data speed of 54 Mbps. The specification uses DSSS with Complementary Code Keying (CCK) for data speeds below 20Mbps and OFDM for speeds above 20Mbps. The specification is backward compatible with the 802.11b specification. 802.11g uses the same frequencies as 802.11b and this causes a reduction is 802.11g speed when there's a conflict with an 802.11b devices. Since the specification operates in the 2.4GHz spectrum, it's susceptible to interference from microwaves and other devices that operate in the 2.4GHz frequency band. The 802.11g effective transmission rate is comparable to 802.11b because the higher data rates in 802.11g are susceptible to more interference than 802.11b.

There are two architectural modes of setting up an 802.11 network. One is the Independent Basic Service Set (IBSS) or the ad-hoc networking mode (Figure 1-1), which is when mobile devices establish direct connections with each other without the need for an Access Point (AP).

FIGURE 1-1 AD-HOC CONNECTION MODE

The other mode is the Infrastructure mode in which devices connect to each other and the network via an AP, effectively forming a Basic Service Set (BSS) (Figure 1-2).

When two or more BSSs are setup to form a subnet, they are called an Extended Service Set (ESS). This is the mode that forms the architectural basis for PALs, where different devices communicate through an AP.

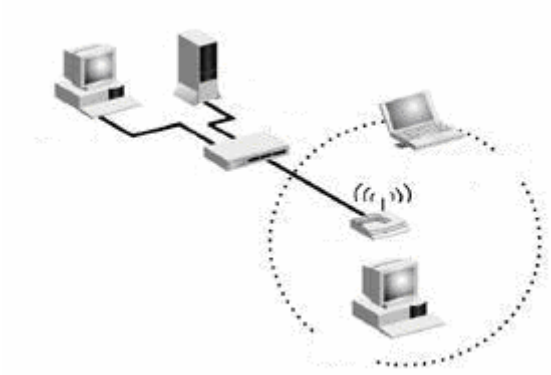

FIGURE 1-2 INFRASTRUCTURE CONNECTION MODE

1.1.2.5 IEEE 802.16

IEEE 802.16 is known as Worldwide Interoperability for Microwave Access (WiMAX). It is a Wireless Metropolitan Area Network (WMAN) standard that allows up to 50km No Line Of Sight (NLOS) communication. This standard operates in the 2-11 GHz and 10-66 GHz (licensed and unlicensed respectively) frequency bands and provides raw data speeds of up to 70Mbps. 802.16 specifies a Physical layer and Transmission layer that correspond to the PHY layer of the OSI model and a MAC layer and Convergence layer corresponding to the data link layer of the OSI model. The MAC layer supports a range of PHY layer specifications. In the wireless networking plethora of technologies, WiMAX stands to provide the 'last mile' access to the service providers and to provide a wireless extension to cable and DSL (Chao L., 2004).

1.1.3 The ETSI BRAN standards

Besides the wireless technologies developed by IEEE, ETSI has also released a group of wireless networking standards under the project named Broadband Radio Access Networks (BRAN). The following are some of the standards that have been developed under BRAN.

1.1.3.1 HIPERLAN2

HIPERLAN2 (High Performance Radio Local Area Network) is a mobile broadband short-range access radio network specification that delivers data speeds of up to 54Mbps at PHY layer. It allows internetworking with other networks including IEEE Ethernet, IEEE 1394 and ATM (Friedrichs B., 2005). This standard was developed in close cooperation with the IEEE WiFi working group and Working Group High Speed Wireless Access Networks (from Japan), to harmonize the operation of standards in the 5GHz frequency band.

1.1.3.2 HIPERACCESS

HIPERACCESS is a radio standard for broadband multimedia fixed wireless access. The standard is aimed at high frequency bands (40.5 GHz to 43.5 GHz) and provides a specification for the PHY layer of the standard. The standard allows for interoperability with the IEEE 802.16 standard (Friedrichs B., 2005).

1.1.3.3 HIPERMAN

HIPERMAN is also a broadband fixed wireless standard aimed to operate at the 2GHz to 11GHz frequency range. The specification is designed to utilize the basic MAC specification of the 802.16 standard. This standard has been developed in close cooperation with IEEE 802.16 work group to enable a subset of the 802.16 specification to seamlessly interoperate with the HIPERMAN specification. The standard offers some of the following service categories; full Quality of Service (QoS), fast connection control management, strong security and capability to operate at non-line-of-sight (Friedrichs B., 2005).

1.1.4 WiBro

WiBro is a broadband wireless technology that operates in the 2.3GHz frequency band. This standard is developed by Korean telecoms industry and is aimed to provide a throughput of 30-50Mbps and to have a 1-5km range of operation. The WiBRO

standard is also developed to ensure interoperability with the IEEE 802.16 standard (Lawton G., 2005).

1.2 Wireless technologies and PAL service provisioning

The available wireless protocols differ in terms of features (Table 1-1) and applicability in PAL service provisioning. Currently the standard that is extensively implemented on PALs, hence the focus of this research, is the WiFi standard (Lehr W., 2004) and this is because:

- The WiFi standard operates in the ISM Radio Frequency (RF) bands. The ISM bands are reserved for usage by Industrial, Scientific and Medical equipment and therefore there are no licensing costs associated with setting up a WiFi PAL.
- WiFi is a data transmission protocol and supports the provisioning of IP based services.
- WiFi is a low-power protocol which makes it applicable for low powered devices that are used at PALs (e.g., laptops, PDAs and tabletPCs).
- WiFi is a LAN (short range) standard and this makes it suitable in meeting the range requirements which are characteristic of most implementations of PALs.
- Device manufacturers are shipping devices with WiFi chips and this has created a demand for WiFi service provisioning.

TABLE 1-1WIRELESS TECHNOLOGIES FEATURE COMPARISON

Protocol	*Data rates*	*Radio spectrum*	*Area*	*Usage*
GSM	9.6 kbps	900 MHz, 1.8GHz	WAN	Voice, Data
GPRS	115 kbps	0.9, 1.8, 1.9 GHz	WAN	Data
EDGE	384 kbps	0.9, 1.8, 1.9 GHz	WAN	Data
UMTS	2 mbps	1.9 – 2, 2.1 – 2.1 GHz	WAN	Voice, Data
Bluetooth	1 mbps	2.45 GHz ISM	PAN	Data
WiFi – b	**11 mbps**	**2.4 GHz ISM**	**LAN**	**Data**
WiFi – g	**54 mbps**	**2.4 GHz ISM**	**LAN**	**Data**
WiFi – a	**54 mbps**	**5 GHz ISM**	**LAN**	**Data**
WiMAX	70 mbps	2 – 11, 10 – 66 GHz	WAN	Data
WiMedia	480 mbps	3.1 – 10.6 GHz	PAN	Data
Zigbee	250 kbps	2.4 GHz ISM	PAN	Data
HIPERLAN	54 mbps	5 GHz	LAN	Data
HIPERACCESS	25 mbps	40.5 – 43.5 GHz	MAN	Data
HIPERMAN	25 mbps	2 – 11 GHz	MAN	Data
WIBRO	50 mbps	2.3 GHz	LAN	Data

The other emerging standard that is gaining popularity as a potential complement to the WiFi standard in PAL service provisioning is the WiMAX standard (Agis et al, 2004). WiMAX operates in both licensed and unlicensed radio frequencies and it promises a wider coverage range and higher data speeds than WiFi. This standard is also set to compete with other 3G cellular technologies in the provisioning of broadband network access.

1.3 Focus on small PALs

PAL service provisioning is an undertaking by different types of entities from municipalities to private owners, large companies to small shops, telecommunications companies to companies in the food industry. Each of these entities presents a different set of characteristics (e.g., business model used, level of technical expertise, size of customer base) that affect and influence the provisioning of the PAL service.

The focus of this research is on small PALs, which are defined formally in Chapter 3. The reasons for focusing on the small PALs are essentially related to their

effectiveness in facilitating further proliferation of wireless computing. Other reasons are as follows:

- The proliferation of large business and their integration into the wider areas of societies is limited. For example, large corporations are based mainly in large cities and metropolitans and, while their density of network coverage in those areas may be excellent, it is normally limited in small and rural towns. Therefore the one way in which the proliferation of wireless network and hence the growth of ubiquitous computing can be facilitated is through the network service being provided by small proprietorships (e.g., coffee shops, book shops, cafes, and libraries) which are already well spread into most geographical areas of society.
- The large portion of current hotspot management solutions and trends in wireless networking favour large businesses implementing these solutions. This is based on the different assumptions on which these solutions are built. The assumptions are about the availability of resources (e.g., technical expertise, hardware, and financial) at the PALs and about the requirements and the characteristics of the individual PALs. For example, there are solutions that assume a specific billing model and hence fail to cater for a different model or free PAL service implementation, some assume a specific level of security needed on the PAL, and others assume the availability of finance to invest in network components.
- The implementation of PALs by smaller businesses provides an environment that facilitates further development and enhancements in terms of new and diverse network service modules being implemented. These also provide a platform for service level customization in order to meet the specific requirements that each PAL faces.
- Availability of a large number of diverse Wireless Internet Service Providers (WISPs) would facilitate and encourage competition between the providers in such a way that the network service users are availed of a wider range of service options from which to choose.

1.4 Research objectives

The primary objective of this research is to derive an architectural framework for small PALs management systems. In view of this, a comprehensive requirements investigation for PALs is undertaken, with a specific focus on small-scale PALs. Small scale is taken to imply the following features, which are best characterised by small restaurants, coffee shops and take-away shops:

- The business is usually a sole proprietorship (this becomes a crucial factor in considering roaming partnerships and agreements).
- The large potion of the customer base is a niche of local, loyal residents.
- The physical area covered by the business is small (within coverage of an AP).
- The first line of the business is not related to wireless network service provisioning (i.e., no dedicated IT expertise in the business).

These requirements will be specific to small scale PAL service provisioning and will constitute the different functional aspects needed to ensure efficiency and effectiveness of the PAL. Some of these functional requirements include:

- Authentication, Authorization and Accounting (AAA) requirements – evaluating the requirements for small PALs with regards to Authentication on the network, Authorization to network services and Accounting for the network usage.
- Billing – this is part of accounting for the network usage. Further evaluation of the available billing and charging models with regards to appropriateness to small PALs.
- Security – determining the level of security that needs to be implemented at a hotspot and juxtaposing that with the effects of increased security on usability and ease of use of the system.
- Network Management – determining the requirements for managing the hotspot.

The main motivation and guiding design principle for the architectural framework will be to provide extensibility and simplicity in the system. Extensibility, in its simplest form, refers to the ability to change and modify the system data and

functionality (Crawford et al, 2003). An extensible system becomes necessary in the light of the need for service level customization and the ability to modify the different aspects of the system to meet the specific customer needs and owner's preferences. As the proliferation of wireless PALs continues, service differentiation will play a main role as a marketing strategy to secure business customers. Extensibility of the PAL management system will therefore become important in facilitating the implementation of network services that are tailored to the exact preferences and needs of the market niche target by the WISP.

Subsequent to the formalisation of the framework, a system implementation that is based on the derived architectural framework will be reported on. The implementation is undertaken on a proof-of-concept scale to validate the framework against actual implementations of PAL management systems and to test its applicability in meeting the identified requirements for small PALs.

1.5 Thesis Overview

The rest of the thesis is organised as follows:

Chapter 2: highlights the fundamental features and requirements of PALs with an emphasis on their level of importance in a PAL implementation. These are discussed under the broad categories of hardware, Authentication, Authorization and Accounting (AAA), network management, content management and roaming.

Chapter 3: provides a discussion of the specific requirements for small PALs. This discussion is based largely on a questionnaire that was undertaken to determine both the users' and the providers' expectations as far as PAL service provisioning is concerned.

Chapter 4: is a detailed discussion of the proposed architectural framework. This chapter highlights the design considerations for the framework and discusses the building blocks of the architecture.

Chapter 5: Subsequent to the formalisation of the architectural framework, a system was developed that implements the proposed framework. This chapter provides the implementation details of the system.

Chapter 6: is the results and discussion chapter. The applicability of the architectural framework and its efficacy in meeting the requirements for small PALs are discussed. This is done through a juxtaposition of the implemented system with other currently available PAL management systems.

Chapter 7: is a conclusion chapter that highlights the success of the research by assessing the degree to which the research meets its objectives. It also discusses further research that can be undertaken.

2 Chapter 2: Requirements for PALs

Wireless PALs serve the main purpose of proving public broadband network services to mobile users through WLAN (Minoli D., 2002). There is a wide spectrum of different services that can be offered on a wireless PAL ranging from basic HTTP, multimedia services to VPN solutions for access to private networks. The basic setup for a wireless hotspot may simply consist of an AP connected to a LAN, but while that could be a working solution, it lacks a number of different infrastructural considerations that could be implemented in order to improve the service provisioning. Some of these considerations, which will be discussed in detail in later sections, include business concerns for profitability and cost-effectiveness of the service, security concerns, hardware concerns with regard to scalability, and network management.

A discussion on the requirements for wireless hotspots requires a categorization of the requirements on the continuum of 'absolutely necessary' features to 'nice to have' features. Necessary features are those that are indispensable in the implementation of a wireless hotspot and without which the operation of the hotspot is impossible. The latter group of features, 'nice to have', are the optional features that aid the effective and efficient implementation and running of a hotspot. Another way to categorize the hotspot requirements is to use the keywords defined in RFC 2119 to indicate the necessity of each requirement. These words are indicated in Table 2-1 and their interpretation is specified in RFC 2119 (Bradner S., 1997).

TABLE 2-1 RFC 2119 KEYWORDS

MUST	MUST NOT	REQUIRED
SHOULD	SHOULD NOT	OPTIONAL
MAY	MAY NOT	RECOMMENDED
SHALL	SHALL NOT	

Figure 2-1 shows a matrix of the requirements descriptors used in the categorization of small hotspot requirements. The horizontal scale indicates the level of necessity specified by the keywords; to the right are the keywords that specify absolute necessity and to the left are those that specify optional features. On the vertical scale,

the top half indicates keywords that specify required features and the keywords at the bottom half indicate features that are prohibited.

FIGURE 2-1 REQUIREMENTS DESCRIPTORS MATRIX

2.1 Essential RECOMMENDED Hotspot requirement

The proliferation of wireless hotspots has been facilitated, to a large extend, by the ease of setting up hotspots and the demand from customers for the service. The ease of setting up wireless hotspots is due to the fact that the frequency spectrum in which WiFi operates is the unlicensed ISM bands, and also due to the fact that wireless hardware is becoming cheaper and is being integrated into a lot of mobile devices. In contrast to a few other wireless broadband technologies (e.g., GSM, GPRS), which need a licence, WISPs do not need a license in order to start operating, which reduces the start-up costs associated with hotspots. This means that anyone and everyone can set up a hotspot and this, together with the fact that there is no external regulation from the government, places pressure upon each WISP to stay ahead of competition. One of the effective strategies for staying ahead of competition is through service differentiation. While this argument for service differentiation is based on the assumption of a retail business model, *i.e.* where WISPs sell network resource usage directly to consumers, service differentiation is still necessary in order to meet the customer's need accurately. Therefore even if the hotspot is provided as a free service for hotel guests, or for airline passengers, service differentiation would still be necessary in order to accurately and closely cater for the specific needs of that market group or niche.

Service differentiation is defined as developing unique product differences along key features and minor details with the intent to influence demand (Piana V., 2003). Two kinds of differentiation can be identified:

- Vertical differentiation - where services can be ordered according to an objective quality or feature of the service. This works on the customer's perceived difference in quality between services. An example of an objective feature that can influence perceived difference in quality is price for the usage of the service (Piana V., 2003).
- Horizontal differentiation - occurs when products differ according to features that cannot be ordered. This is normally due to customer's individual tastes and preferences (Piana V., 2003).

Service differentiation is facilitated by the ability to customize specific features of the service, in this case wireless hotspots, in order to emphasize those features which would lead to higher perceived quality of the service, or those features which cater for the preferences of the target market group for the WISP. Examples of customizations that could be implemented on hotspots are the following:

- Changing the business model of the hotspot.
- Improving the QoS for the service.
- Implementing a different billing model.
- Implementing an authentication protocol that leverages on the customers existing service accounts. For example, using the .NET passport, or authenticating against an existing third party central server.
- Improving the security offering on the network.

The decision as to which service differentiation strategy should be implemented is one that rests entirely with the WISP. Therefore each individual WISP should be empowered enough to make the necessary customizations to the hotspot management system. This necessitates that the hotspot management system exhibits features and characteristics that would enable the WISP to easily and effectively implement the desired changes to the system.

The *RECOMMENDED* features for hotspots management systems therefore are simplicity and extensibility. Extensibility is simply the ability to extend a program, in terms of both the data and the functionality of the program. These two features are normally inversely related *i.e.* the more extensibility features are implemented in a system, the more complicated the system tends to become (Allen E. E., 2001). A balance therefore has to be maintained between allowing for extensibility hooks into the system and keeping the system as simple as possible.

2.2 Hardware infrastructure

Setting up a wireless hotspot requires getting the appropriate hardware that ultimately enables a customer to get access to the network service offered. There are a number of different hardware setups and physical configurations that can be used depending on different factors e.g. the expected number of users, the level of interference at the location and physical considerations of the venue for the hotspot. Due to the integral differences in the venues for hotspots, an extensive site survey should always precede the actual implementation and installation of the hardware components. The survey *SHOULD* consider the factors related to: physical environment, obstructions, building materials, antenna types and placement and data rates (Minoli D., 2002). The different hardware components that can be used on a hotspot include routers, access points, wireless bridges and repeaters, web servers and AAA servers. (Minoli D., 2002).

Despite the different physical and logical network setups possible, the one essential piece of hardware that *MUST* be there in a hotspot deployment is an Access Point (AP) (Intel, 2003). An AP provides the wireless interface between the mobile device and the WLAN. There are different models for APs depending on the wireless standard supported (e.g., 802.11b AP, or 802.11g AP). They also differ depending on the feature set built into the AP and the level of configurability integrated into the AP. There are also APs that support both the 802.11b and the 802.11g specification (DataLink DWP 2000AP), while others feature tri-band antennae (e.g. HyperGain® Model HG2458CU) that operate at 2.4/5.3/5.8 GHz frequency bands to provide connectivity to 802.11b, 802.11a and 802.15 networking protocols.

2.2.1 Scalability

Scalability, as far as hotspots are concerned, is the ability for the network to adapt to increased demands. This means the ease with which the network can handle more customers or higher traffic demands. The hardware infrastructure implemented at a hotspot *SHOULD* be able to be extended (e.g., via adding more components), for example, more wireless repeaters or wireless bridges, in order to cater for the changing demands on the network.

2.3 Authentication infrastructure

Authentication serves the purpose of identifying the users on the network i.e. validating the identity of the users. This is normally done via a username/password pair that uniquely identifies the network users and it is usually assumed that knowledge of the password is a valid means of identifying a specific user. Authentication on hotspots is based on the Universal Access Method (UAM) mechanism which provides a web portal login (Intel, 2003). While a basic username and password pair is sometime adequate for authentication, there is always the possibility of passwords being stolen, or revealed by accident. Other authentication mechanisms that can be implemented at a hotspot include:

- Authentication via global database servers. This builds on the idea of increasing the simplicity of using a hotspots service by allowing the users to logon to the hotspot with their existing credentials i.e. msn.com, aol or yahoo login credentials. This method effectively eliminates the need to register at every hotspot service that the user wants to use.
- Third party authentication is implemented by using systems that handle the user authentication on behalf of the hotspot service.
- Hardware authentication is implemented at a device level in a similar manner that GSM networks authenticate cellular phones. This method is facilitated by the usage of a SIM module that contains the authentication credentials of the users (Balachandran et al, 2005).

The main reasons why authentication measures *SHOULD* be implemented on a hotspot is for the purposes of controlling and limiting access to the network resources

usage and being able to identify the users on the network for Authentication Authorization and Accounting (AAA) purposes. Authentication infrastructure also facilitates non-repudiation, which is a feature that is becoming crucial in the light of business transactions that are carried on the internet (McCullagh et al, 2000).

The underlying protocol for wireless hotspots, 802.11, defines two ways of authenticating devices on the network. One is open system authentication (OSA), where every device that needs connection to the 802.11 network gets accepted without any need for authentication. The other authentication mechanism is shared key authentication (SKA) where the access point issues a challenge-response with the device based on a pre-shared key. This is authentication at the physical layer of the hotspot. Authentication SHOULD be implemented at the application layer as well, in order to be able to identify the users on the network and not so much the devices.

2.3.1 Different authentication methods

A number of different authentication methods and protocols exist that can be used on a hotspot. These range from the simple authentication against a database or using the more advanced authentication against a RADIUS server. As mentioned in the preceding section even more advanced methods of authentication are possible using digital signatures and certification authorities. The level of authentication and the methods implemented are determined by their adequacy to meet the specific need of the particular hotspot. Therefore while a basic database authentication using a username-password pair credentials may suffice for a simple HTTP service at one hotspot, it might not be as adequate for an e-commerce transaction at another hotspot. Some of the protocols and standards that *MAY* be implemented in the authentication process include:

- Challenge Handshake Authentication Protocol (CHAP) is a threeway handshake protocol (Simpson W., 1994).
- Extensible Authentication Protocol (EAP) is used between dial in user to determine what authentication to use (Blunk et al, 1998).
- Password Authentication Protocol (PAP) is a basic username and password authentication system (Lloyd et al, 1992).

- Remote Authentication Dial-In User Service (RADIUS) is a system that allows access servers to communicate with central servers to authenticate users and authorize their access to resources (Rigney et al, 2000).
- S/Key – a one time password authentication system.
- Terminal Access Controller Access Control System (TACACS) (Finseth C., 1993)
- Kerberos - is a third party authentication service used to verify user's identities (Steiner et al, 1988)

2.4 Billing infrastructure

Billing is implemented within the context of accounting for network usage. There are different reasons why billing infrastructure *MAY* be implemented on a wireless hotspot. It is normally used as a means to get a return on investment (ROI), as a means to control congestion, or as a charge for the infrastructure (e.g., reliability, speed, and QoS) provided. The full discussion of the arguments both for and against charging for wireless hotspots is presented thoroughly on the section on billing for hotspots (Section 2.8.1). A flexible, extensible billing infrastructure would nonetheless allow the WISP to make the relevant decision with regards to what business model to adopt for the hotspot.

2.4.1 Different billing schemes

A crucial element that *MUST* be present in a billing infrastructure is a billing scheme. This is simply a method of charging for the network usage, it is the description of the actual calculations that are made to determine the final charge. A number of different billing schemes, each with advantages and disadvantages, and each appropriately applicable in different scenarios, can be implemented. Some of these schemes include, flat-rate pricing scheme, usage-based pricing scheme and expected capacity charging scheme (Section 2.8.2). The functionality that *SHOULD* be provided by the billing framework is that of allowing for the billing scheme implemented to be changed and modified in order to meet the requirement of the particular hotspot. The functionality would also aid in the effective facilitation of service differentiation because different services would inherently need different billing schemes to be applied.

2.5 Network management framework

The network management framework is necessary to ensure an efficient and effective operation of the wireless hotspot. Tasks in network management include the initialization and the configuration of the associated network components, controlling and monitoring of the network, managing the users on the network and implementing the required security infrastructure on the network. Network management infrastructure *SHOULD* be implemented on wireless hotspots to ensure that the available resources are used as efficiently as possible. Part of implementing the features afforded through the extensibility hooks involves having a management framework in place.

2.5.1 Nomadic usage of WiFi hotspots

The proliferation of wireless networking has greatly facilitated the emergence of technomads; people who use communication systems and portable devices to connect to data and internet at their home or work networks (SearchNetworking, 2003). This functionality is also sometimes referred to as transparent virtual networking. The bulk of functionality needed by nomadic users is encapsulated and afforded by the roaming infrastructure implemented. In order to facilitate usage and to enable nomadic users on the network, the following are a few of the factors that SHOULD be taken into consideration (Minoli D., 2002):

- Dynamic Host Configuration Protocol (DHCP) – this is an internet protocol for automatic configuration of computers that run TCP/IP. This protocol MAY therefore be used for configuration of wireless clients on a wireless hotspot. When a client is configured to use DHCP, it can be configured on any wireless LAN that has a DHCP server. DHCP eases the task of configuring a large number of clients that come onto a network.
- Network Address Translation (NAT) – simply allows a number of different IP addresses to access the internet via one IP address. The main advantage afforded by NAT is the ease of managing the network. NAT also facilitates implementation of security measure on the network because it allows for

connections to be easily initiated from within the network and by default indirectly restricts access initiated from outside the local network.

- Service discovery protocol – the nature of nomadic usage is such that the mobile devices do not know beforehand the state of the network that they are going to attach to. As new and improved services are deployed and implemented, it becomes crucial for the devices to be able to dynamically identify and discover the type of services offered at the particular network. Service discovery protocols *MAY* be implemented on a hotspot depending on the number of different services provided on the hotspot.

2.5.2 Security

The word 'security' in the context of information technology and networking usually raises a connotation of privacy, integrity and non-repudiation. Privacy in simplest terms means that data is read by the audience to whom it is intended and not any other third parties. Integrity is the property of the data being accurate and not having been modified between the sender and the receiver. Non-repudiation is a feature that has become crucial on networks especially for e-commerce services, which ensures that the communicating parties are who they are i.e. validating identity, and that neither can deny having undertaken the transactions. Security also raises the idea of the validity of the identity of the communicating parties (i.e., adequate authentication of the parties). Adequate security measure *SHOULD* be implemented at WLANs to ensure an appropriate level of privacy and integrity for the particular type of service and also to balance the cost of a security breach with the cost of implementing a security feature.

A large number of wireless security breaches are a result of 802.11 wireless networks that are deployed with insufficient security measures in place. This is reiterated in the following statement that was released by Gartner (Chandra P., 2002).

"By the end of 2002, 30 percent of all enterprises will risk security breaches because they've deployed 802.11b wireless local area networks (WLANs) without proper security."

2.5.2.1 WEP

WEP is used on WLAN to authenticate radio Network Interface Cards (NICs) to an AP and to ensure privacy of data that is communicated between the AP and the client. The protocol allows for a one-way authentication of NICs on APs but it doesn't allow for mutual-authentication i.e. validation of mobile devices to APs and AP to mobile devices (Geier J., 2002).

The WEP protocol has been shown to be insecure from successful attacks that were carried out to exploit the vulnerabilities in the protocol. Some of the attacks carried out by Borisov et al included passive attacks to decrypt traffic and active attacks to inject traffic (Borisov et al, 2001). The main vulnerability is due to the static nature of the keys and the short length (24-bits) of the Initialization Vector (IV) which, for a very busy AP, leads to the same key being used for more that one packet of data.

2.5.2.2 802.11 TGi

The 802.11 Task Group I has, since the vulnerabilities in WEP were exposed, been working on a security protocol that would replace the WEP protocol. The interim solution was Temporary Key Integrity Protocol (TKIP) which replaced the 24 bit keys that are used in WEP with larger keys (64-bits and 128-bits) (Fleishman G., 2002).

The other security standard for use on WLAN is 802.1x which is simply an authentication framework that can be used with a number of different authentication protocols e.g. Lightweight Extensible Authentication Protocol (LEAP), Extensible Authentication Protocol-Transport Layer Security (EAP-TLS) or Remote Authentication Dial-In User Service (RADIUS). 802.1x provides port-based access control to mobile devices and mutual authentication between mobile devices and APs. The issue with using some of the proprietary protocols, such as Cisco's LEAP is that they are not flexible enough in terms of interoperability (Geier J., 2003).

In late 2002, the WiFi alliance released a new standard called WiFi Protected Access (WPA). The WPA protocol includes both the TKIP and the 802.1x authentication

mechanism which provide mutual authentication and dynamic key encryption. Since WPA implements 802.1x, it can be interfaced with other authentication protocols such as RADIUS and EAP (Linderman et al, 2004).

The ultimate security standard for WLAN developed by the task group is 802.11i a.k.a. WPA2. This standard uses the Advanced Encryption Standard (AES) and includes Wireless Robust Authentication Protocol (WRAP) as the authentication protocol and Counter-Mode CBC-Mac Protocol (CCMP). Due to the CPU-intensive computational requirements for implementing AES, hardware upgrades and replacement are a must for implementing the WPA2 standard.

2.5.3 Quality of Service (QoS)

Quality of Service is the term that is used to refer to level of guarantees in terms of the performance (e.g., uptime, throughput, latency, and error rate) results that can be delivered on the network (Mitchell B., 2005). This is normally achieved by increasing the priority of one service on the network and reducing the priority of another service using congestion-management mechanisms. The 802.11 Task Group E is responsible for implementing QoS at the protocol level, i.e. coordinating between AP and devices to give different priorities to packets that are communicated between them (Mangold et al, 2002). QoS measures *MAY* be implemented on a wireless hotspot to provide the needed performance guarantees for different services. QoS also becomes essential when seen in the light of billing and charging for the network usage. For example a WISP could use QoS as a means of service differentiation on its hotspot or QoS could be used to justify different pricing packages.

In a discussion of QoS in WiFi hotspots it is essential to understand the nature of the underlying communication medium and the factors which could compromise the level of QoS on the network. Since 802.11 operates in the unlicensed ISM bands, there is a possibility of interference from other devices that operate at the same frequencies e.g. microwaves and other access points.

2.5.4 Network usage auditing

The Accounting requirement of the AAA functionality involves collecting data about network usage for purposes of trend analysis and capacity planning, billing, auditing and cost allocation. The information that is collected includes session statistics and network usage data. Network usage accounting and auditing serves the main purpose of controlling, monitoring and aiding in decision making and hence is a functionality that *MAY* be implemented at hotspots to ensure effective and efficient operation of a wireless hotspot.

2.6 Content management framework

The management of the content that is made available is a functionality that *MAY* be implemented on a hotspot to facilitate and to enhance user's experience. The scope of the content management framework would be to determine what content is displayed, how content is displayed and other factors that affect the content displayed to users. For example, content filtering functionality can be implemented by a WISP to block out websites with undesirable content. The other functionality that could be provided within the content management framework is web proxying; which improves performance by caching web pages, and website mirroring.

2.7 Roaming infrastructure

Roaming is the ability for a user to use multiple WISPs while maintaining a formal customer-vendor relationship with only one WISP (Bridgewater, 2003). Roaming is a functionality that *MAY* be implemented to improve customers experience when using hotspot services. It allows a customer the fullness of mobility with the ease of paying for the network usage. A roaming infrastructure should offer measures for the management of the business relationships between the WISPs, it should take into consideration the issue of trust and security when communicating transactions data between WISPs, it should address the issue of compatibility between the systems used by the different WISPs and it should also be able to be improved and extended to accommodate the new improvements in roaming standards.

Different protocols exist that seek to establish a standard way for facilitating roaming between service providers. These protocols define the methods and the procedures that must be adhered to in a roaming transaction. For example they define the data types that should be used, the transport protocols to be used, encoding techniques for all the data that is communicated between the roaming partners. Some of these protocols also define the business aspects related to settling the bills between the involved service providers. Some of the currently defined roaming protocols include:

- BARG – Billing Accounting Roaming Group
- TADIG - Transferred Account Data Interchange Group
- TAP – Transferred Account Procedure
- IOT – Inter Operator Tariff
- GRX – GPRS Roaming eXchange
- CIBER – Cellular Inter-carrier Billing Exchange Roamer

2.8 PALs Network Services Billing

The uptake of wireless networking service provisioning by companies is fuelled by the potential profitability and financial benefits that can be derived from offering the service. One aspect of hotspot service provisioning that greatly determines its profitability is its billing infrastructure. Charging and billing serve the purpose of recovering the investment costs, generating profit for the shareholders and the WISPs and it can also be used as a means of regulation and counteracting congestion on networks (Scalise K., 1999)

In an attempt to answer the question of how to best price internet services, different proposals have been put forward (Mason R., 2000):

- Non-pricing mechanisms – this proposal suggests that all TCP unfriendly traffic should be restricted and controlled on the network. One of these mechanisms was suggested by (Floyd et al, 1997).
- "First Best" pricing mechanisms – in this mechanism, the individual packets bid separately for priority on the network. This mechanism was proposed by Mackie-Mason et al in a paper title "Some Economies of the Internet".
- "Second Best" pricing mechanisms – this model suggests charging for the network based on the demand for different service classes. The basis of this

proposal is the application of price discrimination in charging for network services. This mechanism was suggested in (Odlyzko A., 2003).

Different charging models and methods exist in the telecommunications industry and most of these are derived from the models that were used on circuit switched voice networks (Da Silva R., 2000). The introduction of packet switched services on the cellular networks (e.g. SMS, WAP, GPRS) prompted for a revision of the billing models that were used on the GSM networks in order to accommodate for the difference in charging metrics, i.e. while GSM models charge for time and the distance that the call is made for, GPRS models charge for the amount of data that is transmitted on the network. Besides the difference in the metrics, the actual cost of providing a packet switched service is different from that of circuit switched services. Analysts have calculated that the cost per bit in packet switched services is 2% of that for circuit switched services (Mutooni P., 2000).

In order for a model to be appropriate and relevant to a network service, it has to be balanced between different constraints. Reichl P *et al* (1999) describe this situation as "the feasibility problem of the internet" and they determine that each pricing model has to balance out three requirements; technical feasibility, economic feasibility and user acceptance (Reichl et al, 2001). The resulting optimal model (Figure 2-2) is then the appropriate model to be implemented for pricing the specific service.

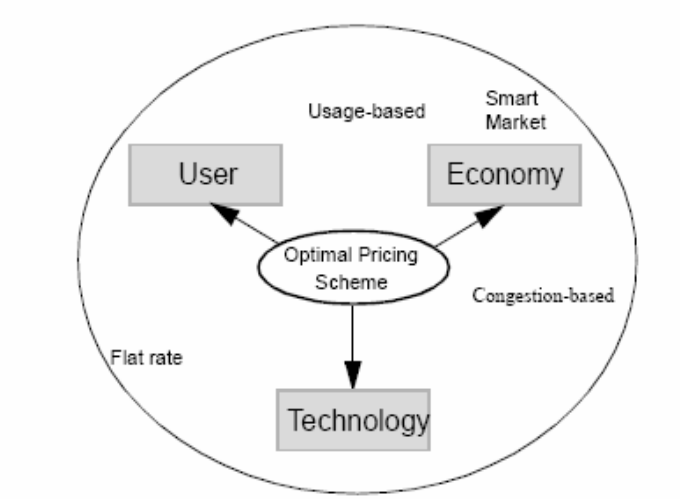

FIGURE 2-2 THE SPACE OF PRICING SCHEMES (REICHL ET AL, 1999)

2.8.1 Hotspot billing considerations

Different considerations and factors have to be taken into account when implementing a WiFi hotspot service. These considerations reflect on the different issues, some economic in nature and others as a matter of usability - in terms of convenience to the user, that influence the successfulness of a hotspot implementation.

2.8.1.1 Free WiFi hotspots

The Internet is a global network connecting millions of computers for exchange of data, news and opinions and as a result the vast majority of information available on the Internet is free of charge. Alongside this exchange of data and information, the internet also facilitates commerce via e-commerce services (e.g., online shopping, internet banking). The free availability and exchange of information on the internet doesn't nonetheless say anything about the value of the information that is provided. The value of a commodity, and for that matter the value of anything, using the principle of demand and supply is conferred by the consumer seeking that commodity (Ballvé F., 1956). The free availability of information on the internet forms the basis for the arguments against charging for PAL service provisioning. The proponents of free WiFi argue that WiFi should not be charged also because:

- The ISM bands in which WiFi operates are free
- ISM bands are inherently unreliable and there's an upper limit to the speed on the wireless networks. This is normally a counter to the argument that WiFi is charged because of the reliability and the speed that is offered.
- Free WiFi is desirable to many users.

Further suggestions put forth by free WiFi proponents with regard to the billing for WiFi hotspot include the following:

- WiFi should be provided as a freebie or an amenity and used as a marketing strategy to secure a competitive advantage for the provider. WiFi is not a stand-alone business, rather an add-on to other communications business (Needleman R., 2005).

- If offering WiFi is relatively cheaper than the main product or service offered, then it should be provided free of charge. (e.g., free WiFi for the international airline passengers.)
- Other revenue channels for WiFi hotspots should be explored. An example would be to finance WiFi from advertisements or have a company sponsor a WiFi hotspot for marketing purposes.

Despite the arguments between pro-fee versus pro-free WiFi proponents, one thing remains clear and that is the fact that the traditional business model of charging for internet is not always appropriate for WiFi hotspots (Cook J., 2004). Other business models that are being explored for WiFi include wholesale business model where providers charge for setting up WiFi networks at different venues while the owner of the venue administers the distribution of the resource, either free or for a fee. This is a model that is used by Wayport Inc. (Meisner J. et al, 2004). WiFi is a great technology that is in search of a market. The WiFi market is also in search of a business model and without doubt the business considerations of implementing WiFi, rather than the technological drive, is what is holding back extensive deployment and proliferation of WiFi hotspot (Deans H., 2004).

2.8.1.2 Network usage payment models

Current methods of paying for network usage include pre-paid and post-paid charging (Sherlekar, 2003). The former method, with regards to WiFi hotspots, can be implemented via a coupon system, where the user buys a coupon and the amount due for the usage gets deducted from the coupon, or the customer can use the system and settle the bill in a similar manner to settling a restaurant bill. The latter method is facilitated by maintaining an account for each user and crediting the respective accounts with the amount due for the resource usage. Pre-paid tariffs are normally less complex than post-paid tariffs which are based on a number of different network usage factors. Traditionally, most operators have viewed post-paid customers as the primary customer group and the pre-paid customers have been viewed as the secondary customer group (Kokko J., 2003). This is due in part to the fact that the Average Revenue per User (ARPU) is relatively lower for pre-paid customers than for post-paid customers. Post-paid charging is preferable and ideal for the situation where

the user of the system is a permanent customer i.e. always comes to the same provider for using the service, or resides in the locality of the service provider. These two methods of charging for network usage are also used on GSM networks, where cell phone users can either be charged on an account or where they opt for the immediate payment for the usage of the service (e.g., the pay-as-you-go option on MTN[3]).

2.8.1.3 One bill roaming model agreements

With the proliferation of mobile devices, the need often arises for users to attach to any service provider convenient to their current location (Minoli D., 2002) (e.g., when travelling to a different location where there's no branch of the customer's home provider). While this situation can be remedied by opting for the prepaid payment option, it is still desirable to have the charges included in their monthly bill from their home provider. This roaming arrangement is implemented quite successfully by cellular telephone providers, where they have roaming agreements established between the providers to allow subscribers from one provider to use the services of the other provider. Roaming is formally defined as the ability to use multiple service providers, while maintaining a formal, customer-vendor relationship with only one (Aboba et al, 1999). Roaming agreements are facilitated by roaming protocols which define how roaming clients should be authenticated, what level of authorization should be given to them and how to perform the associated accounting functionality. A roaming architecture or framework that caters for the above mentioned AAA requirement therefore facilitates the mobility of clients while maintaining integrity of the customer-vendor relationship between the client and the home service provider.

2.8.2 Billing Models used in other network services

Billing models are implemented within the context of the networking service that is being provided. The network service in its simplest form is an exchange of content between a client and a content provider. Da Silva identifies three domains that make up a network service in particular mobile network services (Da Silva R., 2000):

[3] MTN is a South African cellular telephone provider

- Mobility Domain - this is the domain that covers the areas related to the position or the location of the client. This information can be used along with the metrics from other domains to provide a service specific to the location of the client and to charge for that service.
- Connectivity Domain - this is also known as the bit-pipe domain and is concerned with the actual delivery of data or transactions between the operator's core packet network and the content source. This domain also caters for the duration of the service delivery and any roaming requirements between the different providers.
- Content Domain - The content domain is responsible for the content that is offered to the clients and this is perhaps by far the domain with the greatest challenge for the operator because some of the content and services that are implemented are created by third parties.

A combination of the different chargeable elements within each of these domains creates a service that is differentiated from other services. The model therefore allows for some flexibility in terms of grouping the different elements for different services (e.g., GSM, GPRS, and WiFi) and integrating them within the overall billing/charging architecture of the service provider.

2.8.2.1 Charging/billing models for GSM

The diverse service implementations in the telecommunication industry especially in cellular telephony have resulted in various billing models being developed and adopted to support the billing infrastructure in those networks. The following are some of the models used in GSM networks:

- Metered charging - in this model the subscriber is charged a monthly connection fee and then for a metered usage of the service.
- Fixed price charging: a fixed monthly charge is levied on the subscriber and then all the local calls are free of charge and long-distance calls are charged on metered usage.

- Packet charging - this is the model that is used extensively with packet switched networks. It involves capturing and counting data packets that are exchanged in a session and charging based on amount of data exchanged.
- Expected capacity charging - this model involved determining the level of network capacity for a subscriber under congested conditions and then based on the subscriber's usage profile, to charge an agreed upon price for that level of service.
- Edge pricing - in this model the subscribers are charged for the network usage along the edges of the network scope and not along the expected path. Edge pricing is more about the capturing of the charging information. Once captured, the information can be used with any other charging model.
- Paris-Metro charging - this method is based on providing differentiated levels of service to the subscribers and charging the corresponding amount for that service.
- Market based reservation charging - this model, attributed to Mackie-Mason, is based on the notion of simulating a public auction of network bandwidth and services. The subscribers place monetary bids that determine the level of service that they get in terms of QoS.

2.8.2.2 Charging/billing models for wired networks

The two pricing models that are used predominantly in the internet based services are flat-rate pricing model and usage based pricing plan.

2.8.2.2.1 Flat rate pricing

In flat rate pricing, the customers are charged a fixed price for unlimited access to the network per period of time (Reichl et al, 1999).

Advantages

- Flat rate pricing is very easy to implement. It does not impose any extra technical requirements on the service provider and its computation is very simple.
- Flat rate pricing has a high level of customer acceptance (Scalise K., 1999).

- It has reduced administrative and implementation costs. (Hernandez C. E., 2000)
- It enables accurate planning and budgeting for the service provider since the amount that is going to be charged is known beforehand.
- The flat rate pricing model has already been in use in the telecommunication industry and so there is a general high level of support and uptake from the telecommunication industry.
- Flat rate pricing encourages the proliferation of internet usage and e-commerce since customers are given an incentive to spend more time on the internet.

Disadvantages

The main disadvantages of flat rate pricing model as outline by Hernedez C. E (Hernandez C. E., 2000) are:

- The price in flat rate pricing does not reflect the usage on the network. The person who is using a lot of network resource is charged the same amount as the person who is using fewer network resources.
- The model does not take into consideration the congestion on the network.
- The different network requirements imposed by different network considerations are not taken into consideration in the flat rate pricing model, e.g. media services generally require higher QoS than a simple web browsing service but this difference in the requirement is not reflected in the price that is charged.
- Flat rate pricing exhibits the problem of 'The tragedy of the commons'. This means that since people are not charged for the actual resource usage on the network, they are encouraged to use more and more of the resources and leading to congestion and over usage.
- In flat rate pricing, small users subsidize heavy users because costs are recovered through the connection fees.
- It hinders the implementation of new network services, e.g. there is no incentive for services that require higher bandwidth because the prices would not reflect that increase in the network resource.

2.8.2.2.2 Usage based pricing

Usage based pricing takes into consideration the usage metrics of the user on the network (e.g., time spent or data downloaded). The main role and aim of prices in usage based pricing is to present information to the users about the true cost of their actions on the network so that users can compare the cost and benefit of using the network and hence make informed decisions.

Advantages

Some of the advantages of usage based pricing are:

- It is relatively better than flat-rate pricing at reflecting the accurate usage of the network resources.
- It minimises congestion since it discourages users from spending lots of time on the network.

Disadvantages

- Although per minute pricing is a simpler form of usage based pricing, it has a higher administrative and implementation cost than flat rate pricing.
- User acceptance of the pricing method is lower for per minute pricing than for flat rate pricing.

As a result of the awareness that consumers are willing to change their behaviour based on the charging model used (Scalise K., 1999), models have been developed that capitalize on that behaviour. One of this is the Paris Metro Pricing (PMP) method.

2.8.2.2.3 Paris Metro Pricing Method

In the Paris Metro Pricing (PMP) method, the users are given a choice of the class of service that they want to utilize which differ only in terms of the prices that the user has to pay in order to use that class of service. This leads to a control measure in

which cheaper services classes get more congested than the more expensive service classes (Mason R., 2000).

Advantages

Some of the advantages of the Paris Metro Pricing scheme as identified in (Mason R., 2000) are:

- Simplicity: this is due to the fact that PMP reduces the complexity of network management
- Low efficiency loss
- Congestion control: the model provides an automatic congestion control mechanism as less people chose to go for the more expensive classes of services.

Disadvantages

- This model may not survive in a non competitive equilibrium.
- It places a need on the network to be able to keep track of each user's choice of service class.
- The model does not provide an explicit QoS which may be necessary for certain services requirements.

An extensive evaluation of the different pricing schemes had been undertaken by Falkner et al (2000) in which they evaluated flat pricing, priority pricing, Paris Metro Pricing, smart market pricing, responsive pricing, expected capacity pricing, expected capacity pricing, edge pricing and effective bandwidth pricing (Falkner et al, 2000). The models were evaluated on the basis of compliance with existing technologies, measurement requirements for billing and accounting, support for congestion control, provisioning of QoS guarantees, degree of network efficiency, degree of economic efficiency, impact on social fairness and pricing time frame.

	Flat	PMP	Priority	Smart market	Edge/exp.cap	Resp.	Efficiency bandwidth	PFP
Compliance	IP	IP, VN	IP		ATM, RSVP	ATM, VN	ATM	ATM, IP
Billing Measures	No	No	Yes	Yes	Yes (local)	Yes	Yes (local)	No
Cong. control traffic management	No	Yes (rel.)	Yes (rel.)	Yes (rel.)	Yes (CAC)	Yes	Yes (CAC)	Yes
Individual QoS	No	No	No	No	Yes	Part	Yes	No
Network efficiency	Low	Var.	High	High	High	High	High	High
Economic efficiency	Low	Var.	High	High	Var.	High	High	High
Social fairness	Yes	Yes	No	No	Yes	Yes	Yes	Yes
Time frame	Long	Long	Short	Short	Medium/long	Short	Short	Short

FIGURE 2-3 PRICING SCHEMES FEATURES (FALKNER ET AL, 2000)

The different pricing schemes have different features and characteristics (Figure 2-3) that influence their applicability in different scenarios.

2.9 Summary

This chapter forms a basis of the analysis of the requirements for PALs by documenting the requirement for small PALs. The essential recommended requirements for PALs is identified as the ability to extend and modify the functionality implemented at a PAL and the simplicity of managing it. Further investigation into network usage billing is carried out, with an introduction to the different billing methods and schemes from the telecommunication industry.

3 Chapter 3: Small PAL requirements

In order to define and design network solutions for small PALs, it is crucial that their characteristics and those of the associated users are formalized. Chapter 2 provided an overview of the fundamental features of PALs in terms of the characteristics and the functional requirements that have to be considered for any implementation of a PAL. In this chapter the focus is now on the specific requirements that have to be taken into consideration for small PAL implementations.

In designing solutions for the PALs, two factors need to be well documented and understood (Intel, 2003). The first factor is the expectations of the PAL users, which differ depending on the different conditions in which the hotspot is implemented. The general categories that should be taken consideration include:

- Customer cost expectation - this takes into account the value that the customers place on the network service that they are utilizing and hence the amount they are willing to pay for its usage. The decision of whether to bill for network usage or not is a function of the environment in which the hotspot is implemented. This leads to various billing scenarios for different PALs, with free PALs on one end of the billing continuum and fee based PALs at the other.
- Performance expectation - is normally the users' expectation with regards to the amount of bandwidth available on the hotspot. This is influenced to a large extend by the type of use of the PALs and the number of users expected on the PAL simultaneously. Intel recommends a minimum of 100 kbps per active user on the PAL (Intel, 2003).
- Security expectation - the level of security measures implemented at a PAL should cater for users' expectation as far as privacy, confidentiality and protection from malicious acts are concerned. This also depends on the type of use of the service. A WISP is responsible for ascertaining the level of adequate security needed on the PAL and to balance the provision of that level of security with the associated cost. This could be in terms of the monetary cost of getting the right infrastructure in place or in terms of the usability costs associated with users having the go through more levels of security.

- Availability and reliability expectation - as far as customers are concerned, the PAL service should just work. They expect the service when they need it. The PAL should provide a service that offers an acceptable level of reliability and availability to meet users' expectation. The service should be able to handle an increasing number of users on the network and to handle potential interference issues from neighbouring APs.

The second factor that should be taken into consideration when designing solutions for PALs is the hotspot environment in which the hotspot is implemented. This is determined as well by the type of the hotspot that is being implemented. This should take into account the following factors:

- Physical size - this is the area of coverage of the PAL. This together with the user density helps to determine the number of APs that must be installed on the PAL.
- Number of users - is a factor that determines the utilization level of the APs. There could be a higher user density per AP, even though one AP covers the whole PAL area, which would entail deploying more APs to alleviate the load. An Intel guideline for the reasonable user density is 20-25 users per AP (Intel, 2003).
- Usage models - of the PAL should be determined in order to provide a service that meets the usage requirements of the users. The usage models indicate the type of services and applications that are utilized on the PAL. This information together with the user density information should aid the PAL in determining the total bandwidth requirements of the PAL.

These two factors are explored in the context of small PALs in order to ascertain the specific requirements of these PALs and hence to provide a formal basis on which to implement the network solutions for small PALs.

3.1 Small PALs environmental characteristics

In a study done by Intel, a number of different hotspot implementation environments and scenarios are described. These range from hotspots that are implemented at coffee

shops to hotspots that are implemented at large venues (e.g., convention centers and shopping malls). In their study, different environmental characteristics for small hotspot are highlighted and described (Intel, 2003). Since there is no standard classification for small or large hotspots, these identified factors are used as the defining factors and the basis for the classification of small hotspots and the formalization of their associated requirements:

- The number of simultaneous users is less than, or equal to ten.
- Therefore the maximum user density is ten users per AP.
- The horizontal physical coverage is approximately $140m^2$ and the vertical physical coverage is approximately 1.8m (i.e. within coverage of a single AP)
- The network access may be required both indoors and outdoors.
- The level of security required on the PAL is simple user authentication and no encryption.
- There is potential interference from neighbouring PALs.
- Billing is required for the network usage.

A further study, conducted via questionnaires, was undertaken to consolidate the small PALs characteristics described by Intel and to validate their applicability for small South African businesses.

3.1.1 Outline of the questionnaire

The main aim of the questionnaire was to ascertain the characteristics of small businesses in terms of the different factors that would influence a PAL environment should the businesses implement a PAL service. The questions (see

Appendix D: WISPs questionnaire) can be roughly grouped as follows:

- Infrastructure - aimed at determining if the businesses were interested in setting up a PAL service, the kind of infrastructure they had in place, in terms of computing resources (e.g., number of PCs, availability of a LAN, connection to the Internet, OSes running on their systems), and in-house technical expertise.
- Characteristics - aimed at determining the associated business characteristics in terms of the business industry sector, the number of customers that are handled at the business per period of time, and the physical coverage of the business venue.
- Preferences - aimed at ascertaining the requirements of the business in terms of the factors that are relevant in PAL service provisioning. The questions included the business preference in terms of the level of security desired on the PAL, roaming relationships with other hotspots, desirable billing schemes, flexibility in altering the billing model and the charging methods to be implemented (i.e., pre-paid or post-paid charging)

3.1.2 Business profiles

The businesses among whom the questionnaire was circulated are all businesses in the food industry (i.e., restaurants and coffee shops). These businesses range from small sole proprietorships to large franchise businesses that handle hundreds of clients per week. The main means of revenue for these businesses is food provisioning and therefore a PAL service would be an option that is implemented as an enhancement to the customers' experience and to add value to the currently offered service (i.e., food provisioning). Consequently, the businesses' customers have a primary demand for food that is augmented by the availability of the PAL service. This factor has strong ramifications in terms of the pricing considerations and the billing arrangements for food and PAL provisioning service at the business.

3.1.3 Observations from questionnaire

All of the businesses (5 in total) that participated in the questionnaire were interested in setting up and providing a PAL service for their customers. The provision of the PAL service is seen by these businesses as a value added service offering to their customers, and not as an alternative line of business. The results from the

questionnaires could be categorized into three sections (i.e., infrastructure, characteristics, and preferences) that defined the business profiles:

3.1.3.1 Infrastructure

- All the businesses have at least one personal computer installed on their premises, this usually hosts the business management system used for food provisioning service.
- 80% of these businesses have access to the internet.
- The level of technical expertise available at these businesses divides into two extremes, with 60% having a high level of expertise and 40% have a minimal level of expertise. The minimal level of expertise is defined as the ability to utilize basic computing functions and application software and the high level of expertise means the ability to set-up, configure and utilize both application and system software and the ability to handle various computing problems.
- 80% of the businesses run a Microsoft Windows™ operating system and the rest run a variant of the Linux operating system.

3.1.3.2 Characteristics

- The smallest of these businesses handled on average about 250 customers per week and the largest serves more than 1000 customers per week.
- All of the businesses are in the food industry.

3.1.3.3 Preferences

- The businesses preferences in terms of billing schemes and the charging method were also ascertained in the questionnaire. The three billing schemes that were considered are free internet, flat-rate billing and usage based billing. The most preferred billing model by the businesses is usage based billing (Figure 3-1). In terms of overall performance it is rated at 46 % compared to 29% and 25% for flat rate billing and free internet access, respectively. The reason commonly identified for this is that usage-based billing provides a very accurate matching of the value that the customer derives from the usage of the system and the associated cost for the usage of the system

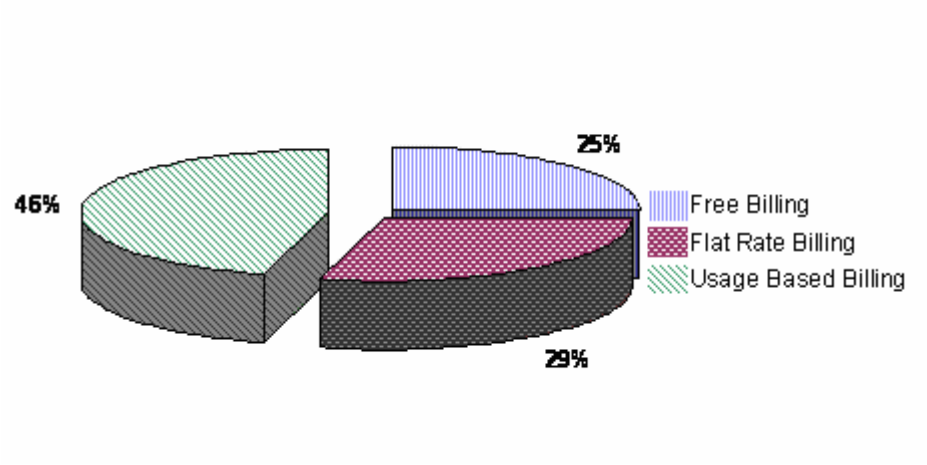

FIGURE 3-1 BILLING SCHEME PREFERENCES

- The two charging options that were considered in the questionnaire are pre-paid and post-paid charging. All the businesses indicated a strong preference towards charging the customers immediately and not on account. This is due in part to the fact that the majority of these businesses do not have formal business relationships with their customers and the normal charging method for the food industry is charging immediately for the food consumed.
- There was a varied perception of the importance of roaming between different hotspots. Forty percent of the businesses ascribed high importance to roaming, while another 40% indicated that roaming was not important at all to take into consideration. Roaming is an option that opens up new possibilities for business partnerships. It is a meaningful arrangement in the context of permanent customer-vendor relationships (i.e., mostly in post-paid charging scenarios which introduce situations where customers from one PAL need to use another PAL service).

These business environment factors cover half the considerations that should be taken into account. The other half of factors to be considered has to do with the customers' characteristics and preferences.

3.2 PAL users characteristics

A study was undertaken to determine the characteristics and the expectations of PAL users. Again a questionnaire survey was conducted (see Appendix E: PAL users

questionnaire), with the aim of ascertaining the users' cost, performance, security and availability expectations of a PAL.

In order to best understand the results of the questionnaire it's important to understand the sample space characteristics; the bulk of the questionnaires were handed out at a national telecommunications conference[4] where a large proportion (48%) (Figure 3-2) of people were in the IT and the telecommunications industry. This user group was targeted as they are the most likely users of PAL services and potentially early adopters of new technologies surrounding PALs. The results of the questionnaire indicated a number of relevant factors that must be taken into consideration when designing network solutions for PALs;

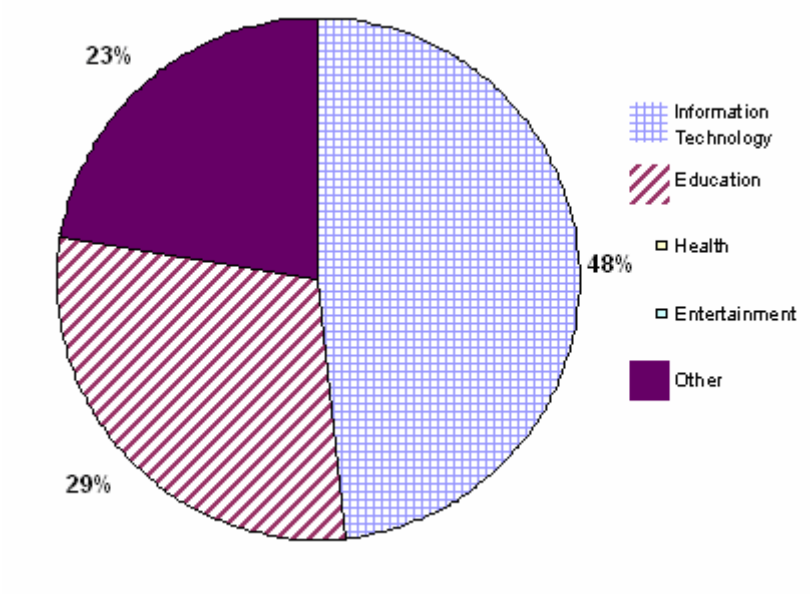

FIGURE 3-2 QUESTIONNAIRE RESPONDENTS' INDUSTRY GROUPS

- A large portion (61%) of the respondents are in possession of a WiFi enabled mobile device (e.g., PDA, laptop, tablet PC) (Figure 3-3).

[4] SATNAC 2004 – see http://www.satnac.org.za

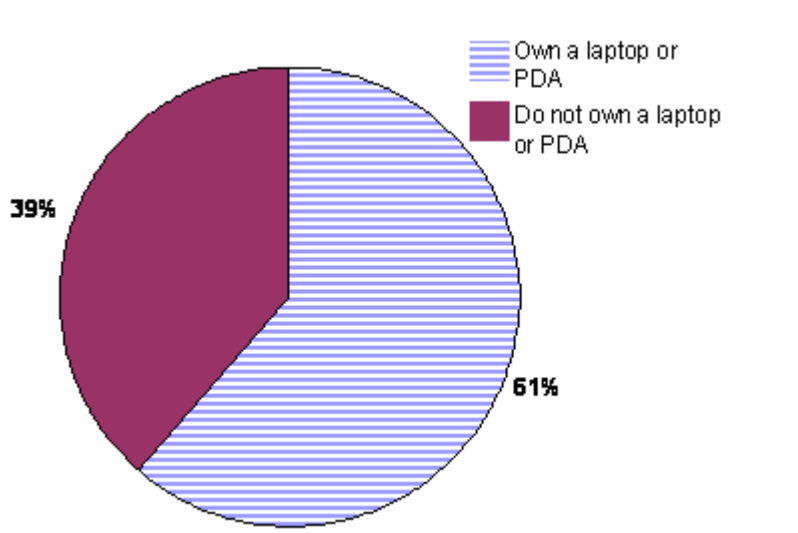

FIGURE 3-3 QUESTIONNAIRE MOBILE DEVICE OWNERSHIP

- In determining the frequency of usage of the Internet, 48% of the respondents indicated an internet usage of more than 40 hours per week (Figure 3-4). Only 10% of the respondents indicated spending 5 hours or less on the Internet per week.

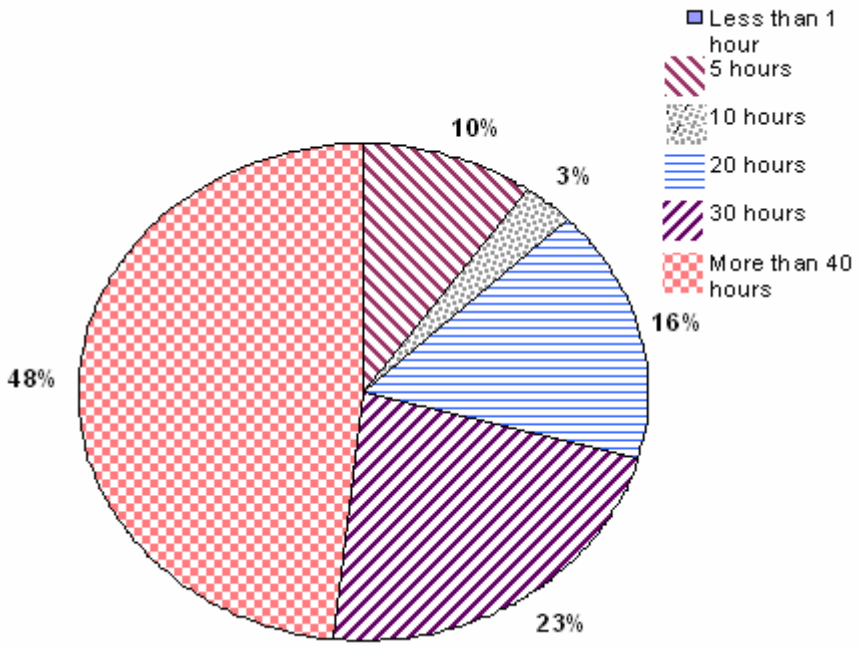

FIGURE 3-4 QUESTIONNAIRE INTERNET USAGE RATE

Seventy one percent of the respondents spent at least 30 hours per week (about 4 hours per day) on the Internet. This becomes an important factor to take into

consideration as it influences capacity planning decisions that have to be undertaken by the WISPs.

- In order to work out the type of services that need to be offered for a PAL, it is important to determine the common and top uses of network access. This helps in determining the different features that have to be implemented on a hotspot. For example, if a large potion of users use the Internet for downloading multimedia files, this directly influences the bandwidth considerations for the PAL. If most people use the internet for services that required a high level of confidentiality, privacy and non-repudiation, this directly dictates the level of security that has to be offered on PAL. To determine this, the respondents were asked to indicate their top three uses of the internet ranked accordingly.

  The top two uses of the internet were e-mail and web browsing at 33% and 30% respectively (Figure 3-5). 9% of the respondents indicated that Instant Messaging (IM) is one of their top three uses of the internet. The proportion of the people who used the internet for connecting to a corporate network was at 6%, this becomes an important factor in determining the need for VPN service provisioning on the PAL. Two percent of the respondents indicated the usage of the network for downloading music files from the Internet.

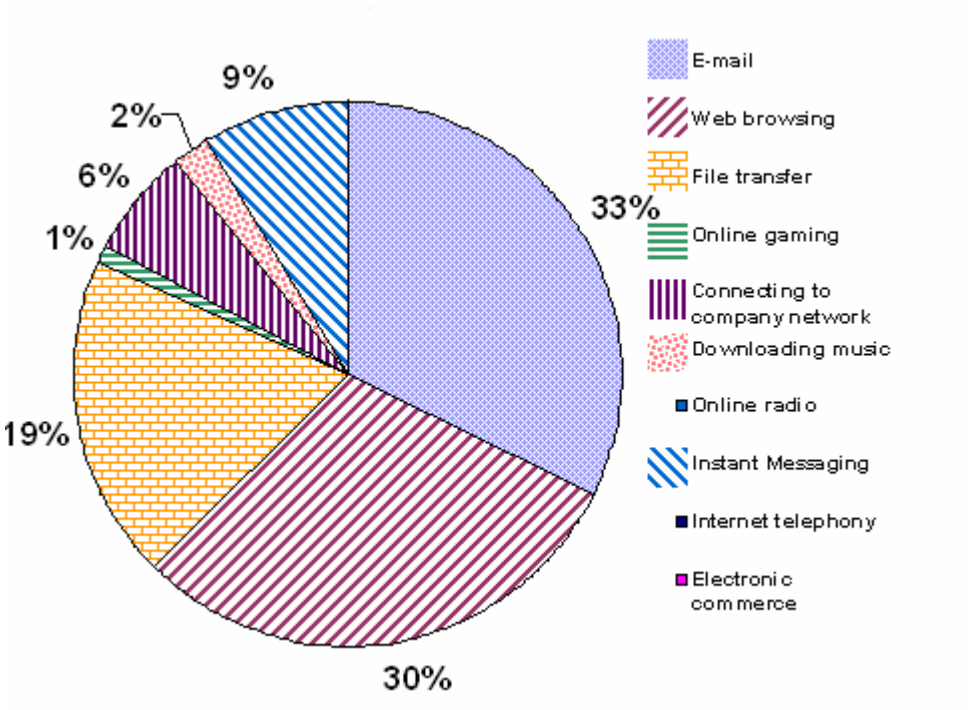

FIGURE 3-5 OVERALL TOP USES OF THE INTERNET

- PAL service provisioning has to take into consideration AAA functionality (Section 2.3). Authentication mechanisms need to be provided on a PAL for the purposes of identifying network users. Authorization is necessary to be able to control access to network services. Accounting ensures that the users' usage of the network is accounted for and subsequently, depending on the business model implemented on the PAL, charged accordingly. Part of the accounting activities on a PAL may include billing and charging for the network usage. Amidst a plethora of billing models and schemes, it becomes important to identify the users' associated preferences.

  The four common billing models that the users were asked to indicate preferences on are (Section 2.8.1):

  - Free PAL usage – where the network access service is provided free of charge
  - Flat rate billing – where users pay a fixed amount per period (e.g., month) and then use the network service as they require.
  - Usage based billing – is where users are charged for the network service based on some usage-related metrics (e.g., the amount of time

they spend on the network, or the amount of data that they communicated on the network).

- Paris Metro Pricing (PMP) – is a kind of a diff-serv (Aimoto et al, 2000) mechanism, in which the network service is segmented into different classes that differ based on the charge levied. The users then choose the service class they want to operate in and get charged accordingly.

The users were asked to indicate which billing model they would desire the most to be implemented at a PAL, and also which billing model is most likely one to be implemented at hotspot. The latter question simply takes into consideration the WISPs' interest in terms of profitability and economic efficiency of running the PAL.

The most desirable billing model came out as the free internet access at 33% followed by flat rate billing at 29%. This pattern in billing model preferences, in particular the preference of flat rate billing over usage based billing, was also documented in a study done at University of California Berkeley (Scalise K., 1999). Usage based billing was desired by only 22% of the respondents (Figure 3-6) while PMP was favoured by 16% of the respondents.

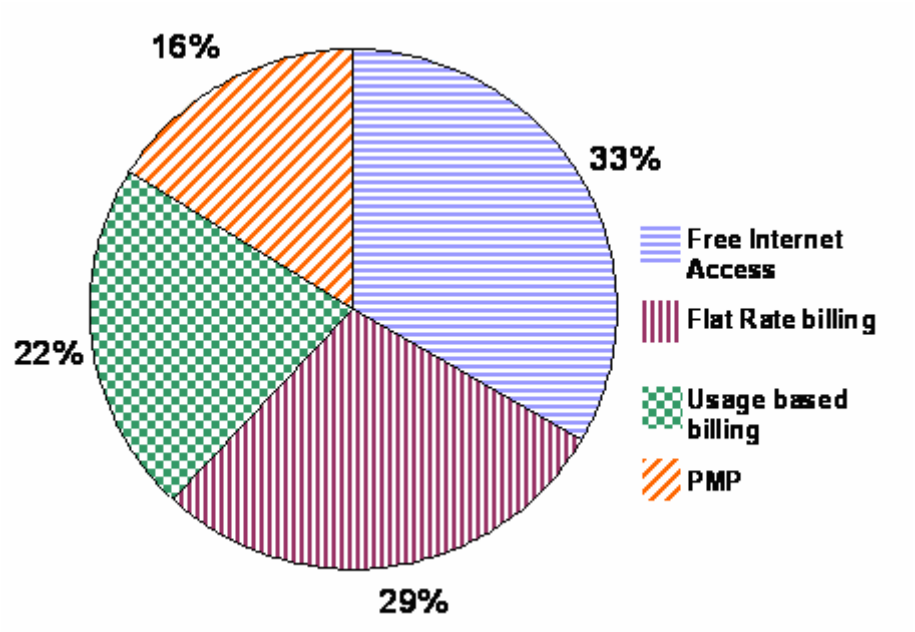

FIGURE 3-6 QUESTIONNAIRE PREFERRED BILLING SCHEMES

When taking into consideration the profitability concerns of the WISPs, the majority (31%) of the respondents indicated that usage based billing is the most reasonable billing model (Figure 3-7). An interesting fact to note is the decline in the 'proponents' of free access in the light of profitability concerns of WISPs. Only 21% of the respondents thought that free access is a likely billing model, as compared to 33% who said they desired free network access. Flat rate billing also decreased, with 29% of the respondents saying that it was a likely billing model. More respondents (22%) thought PMP was a likely model than desired it (16%) as a billing model.

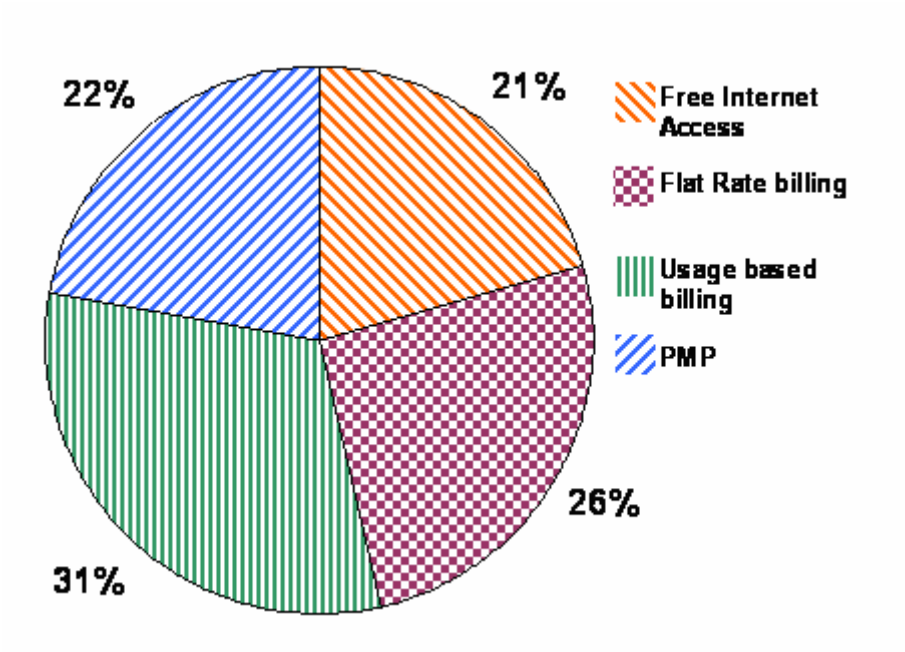

FIGURE 3-7 QUESTIONNAIRE REASONABLE BILLING SCHEME

- The users were then asked to comment on the importance of roaming for PAL users. Roaming is a feature that has thus far been extensively implemented in cellular telephony services. It essentially allows the service provider to offer their customers a better service in terms of the ability to use foreign (implying not home WISPs) providers to access the network service while maintaining only one formal customer-vendor relationship with the home WISP. This feature becomes very crucial in the context of small PALs because as identified in their characteristics, they are mostly single proprietorship

businesses that are not extensively spread to other geographical areas. This means that if the PAL users are permanent customers, they'd benefit from the roaming feature by maintaining a customer-vendor relationship with one WISP while being able to use other PALs for the occasional instance when they are not at their home PALs.

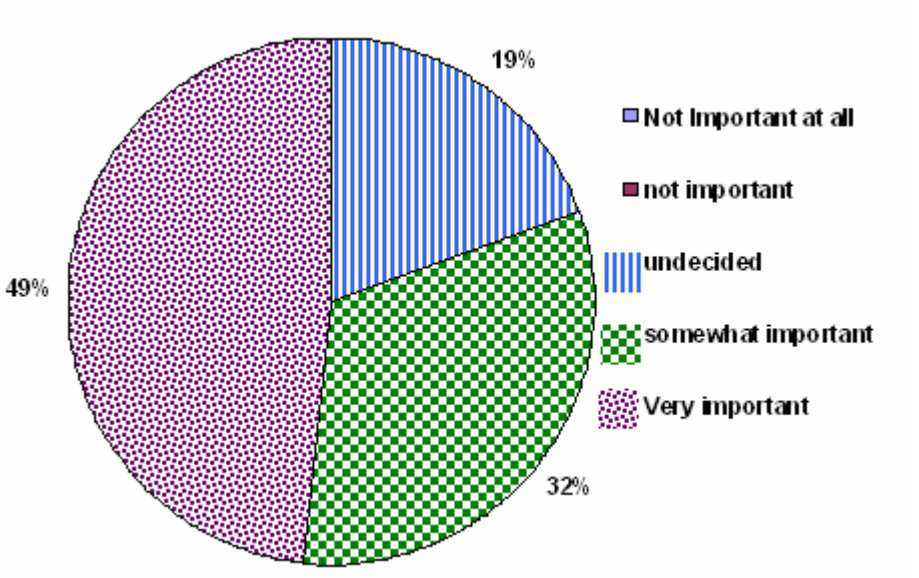

FIGURE 3-8 QUESTIONNAIRE ROAMING RATING

The results from the users indicate a large proportion (81%) of the users are in favour of a roaming facility at a hotspot (Figure 3-8).

- Closely related to the question of roaming is the issue of billing for network usage. The earlier question ascertained the user preference in terms of the billing model that should be implemented at a hotspot. The users were asked to indicate their preferences in terms of how they wish to pay for network usage. The two prominent ways of paying for the network usage are paying on account and paying on use (Section 2.8.1.2).

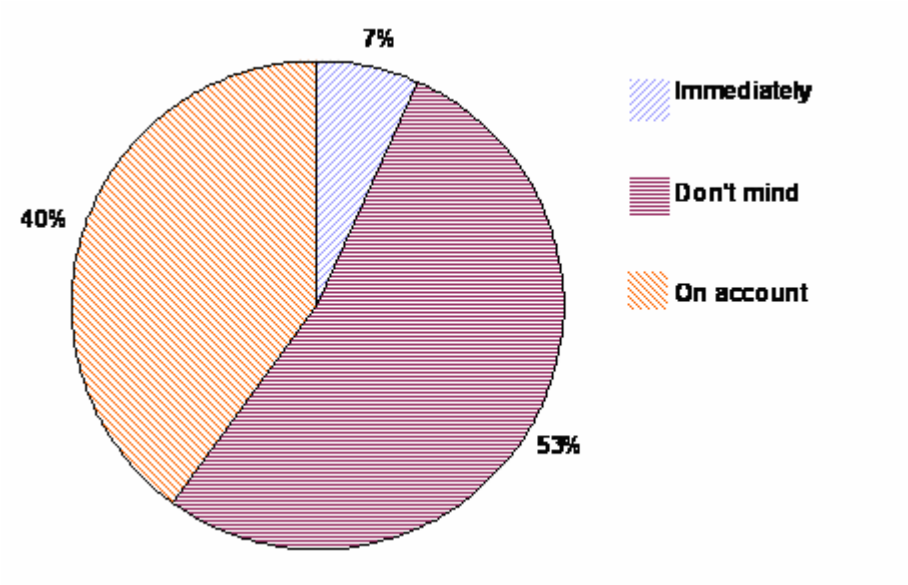

FIGURE 3-9 QUESTIONNAIRE PAYMENT OPTIONS

The majority of the users (53%) were indifferent to the payment method being implemented on a hotspot. Only 7% of the users indicated that they would prefer to pay immediately after use while a substantial 40% said they would prefer to pay on account (Figure 3-9). This question closely relates to the type of payment options that have to be implemented at a PAL. If the majority of users would prefer to be billed on account, then it introduces a number of interesting factors that have to be taken into consideration in the light of roaming relationships with other hotspot providers.

- One of the means of financing a service provisioning undertaking like hotspots is through the use of advertisements. In essence this allows businesses to have their advertisements communicated to the wireless service users. The users were asked to indicate their preferences as far as advertisements that pop up during web browsing are concerned

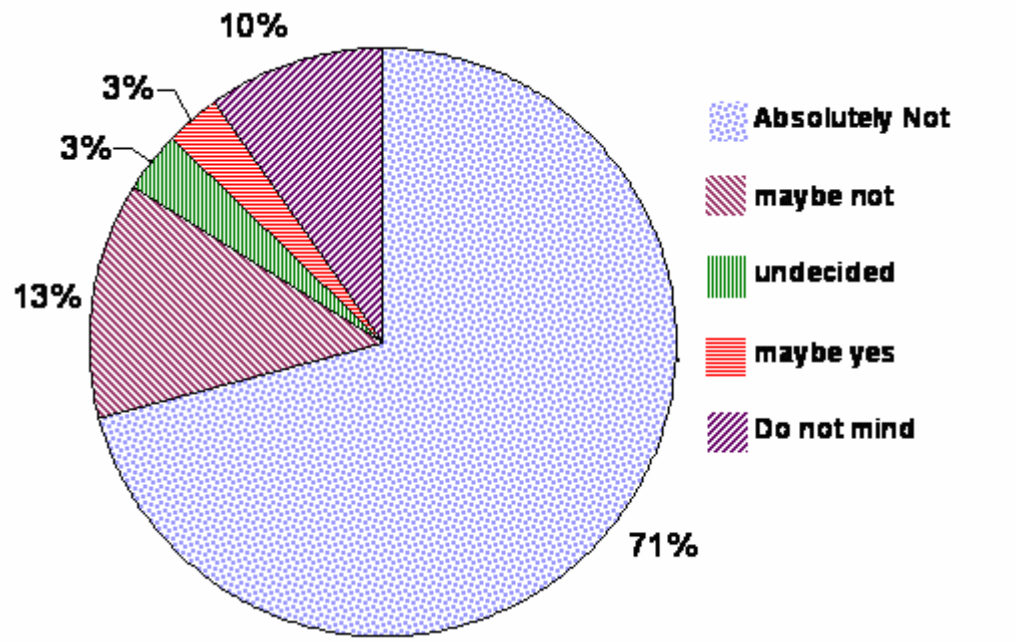

FIGURE 3-10 QUESTIONNAIRE POP-UP ADVERTS PREFERENCE

The vast majority of the users (71%) indicated that they absolutely do not want advertisements during web browsing (Figure 3-10). Ten percent indicated that they do not mind pop-up adverts. This has a direct influence on the billing functionality implemented at a hotspot because it indicates that the WISP cannot for example, finance network service usage by allowing third-party advertisements.

3.3 Observations

The above surveyed features and preferences of both the businesses and the hotspot users are foundational in guiding the design of the network solutions for small PALs. These observations are referenced in Chapter 4 in the context of how they influence the implementation of both the architectural framework and the resultant management system that is implemented in this project. The following conclusions can be drawn from the data collected from both questionnaires.

3.3.1 Network usage billing

There is a seeming conflict in terms of the billing models that the users and the providers prefer (i.e., free internet access is a billing model of preference for the users, while it is the least desired by the providers). On the other hand, usage based billing is

the most desired by the providers while it's only the third preferred model by the users (Figure 3-1 and Figure 3-6 respectively).

The hotspot users recognize, however, that free internet access is the least likely billing model to be implemented at a hotspot taking into consideration the profitability considerations of WISPs. Users also indicated that usage based billing would be the most likely to be implemented at a hotspot.

The seeming conflict therefore balances out at equilibrium where the providers and the users both prefer usage based billing. The fact that there's a fraction of providers who considered free internet access as an option, and the fact that free internet access is the most desired model by users, allows room and flexibility for the providers to be able to alter the implemented billing model at hotspot, say at different times (peak, off-peak, lunch time) or to different customers (once-off customers, customers who buy lunch meals, business clients), or at different occasions (Christmas free usage special, business birthday specials). There is a very limited prospect of financing free internet access from third-party pop-up advertisements as the majority (71%) of the users were strongly against their use.

3.3.2 Paying for network usage

A larger proportion of the users indicated a preference to pay on account than those who specified paying immediately as their preferred option. Fifty three percent of the user did not have any preferences with regards to how they pay for network usage (Figure 3-9). On the other hand, all the small businesses indicated that charging for network usage immediately would be their most preferred method.

This highlights the need for a compromise to be reached between the providers and those users whose preference is for paying on account. This could be in terms of establishing a long term customer-provider relationship with those users. This also highlights the potential ramifications associated with providing a roaming facility between the different hotspots. Roaming makes sense in the light of permanent user-provider relationships**.**

The majority of the users (with 49% choosing ‘very important’) indicated that roaming was an essential functionality to have available (Figure 3-8). The requirement for immediate billing of users almost seems to diminish the relevance of the roaming functionality at small hotspots.

3.4 Other guiding factors

The above mentioned characteristics provide a foundation for the development and implementation of solutions for PALs. Other factors that have to be taken into consideration in the design of the PALs networking solutions include:

- The extensibility of the solution implemented, due to the fact that most businesses are stand alone businesses that would want to plug-in extra functionality modules and to change the system as the need and requirements arise. Extensibility also becomes an issue when taking into consideration the different business model that each business may implement.
- Simplicity of use of the solution is also crucial. This does not necessarily refer to the simplicity of the design of the solution because implementing extensibility mechanisms into the solution almost invariably introduces further complexities into the system.
- It is also important that the solutions are easily integrated into the systems (legacy or not) that are already being implemented by the business.
- A system that relies on technologies that are widely used and standardized would provide a solution that is easy to use, and easy to understand. It would also facilitate the technological support that may be needed for the solution.

3.5 Summary

The defining features of small PALs (e.g., AP density of 10 users, physical coverage of single AP) are outlined based on the study that was conducted by Intel. A questionnaire based profiling of the businesses indicates:

- An availability of an adequate infrastructure (in terms of available computing resources and connectivity to the Internet) with a limited technical expertise.
- A strong preference towards usage based billing scheme and pre-paid billing.

These findings together with the user characteristics outlined below, provide an insight and a guideline in designing the framework for small PALs, discussed in Chapter 4.

- The majority of the users spend at least four hours per day on the internet with the common uses being e-mailing and web browsing.
- Users prefer free internet access on PALs but recognize usage based billing as the reasonable (i.e., taking into consideration the profitability concerns of the provider) and likely billing scheme.
- Post-paid billing is more favoured by the users than pre-paid billing.

4 Chapter 4: Xobogel architectural framework

Wireless service provisioning solutions for small WISPs need to inherently address and take into account the PAL environmental factors and PAL users' characteristics previously identified (Chapter 3). The solutions proposed address these issues at a low architectural level, by defining a framework that is aimed directly at meeting the specific requirements of small WISPs. The architectural framework proposed in this thesis is named Xobogel and this is a play on the word Lego™-Box which encapsulates the extensibility, simplicity and flexibility sentiments and functionality that this framework proposes to provide.

4.1 Xobogel Overview

Xobogel is an architectural framework that facilitates the implementation of small scale, easily deployable and extensible PAL management system. The framework takes into consideration the following factors based on the findings of chapter 3:

- Extensibility - the framework should afford the ability to extend the implemented system's functionality.
- Simplicity and ease of implementation - ease of use and implementation of the framework is necessary in order to facilitate the proliferation of PALs.
- Service differentiation and proliferation - more and more diverse IP-based services are being implemented and developed to meet the diverse customer networking needs. The framework should support the implementation of different network services on the implemented system.
- Business Integration - the framework should allow for both a stand alone implementation and an implementation that easily interfaces with the legacy systems in the business.

4.2 Xobogel Introduction

The Xobogel framework is derived from and designed to take advantage of the currently available technologies and protocols. It therefore builds upon the established knowledge and the experience from the design and implementation of other technologies. The underlying design pattern that was chosen for the architectural basis for Xobogel is the microkernel pattern (Buschmann et al, 1996) and it is interlaced

with the IPDR reference model (IPDR, 2005). The microkernel pattern provides the basic structure that facilitates extensibility, and in turn services differentiation, in the framework. The IPDR specification, on the other hand, provides a means for the exchange of usage data among different service elements in the system thus enabling the integration of the implemented system within the Business Support Systems (BSSs) in the organisation.

4.3 Microkernel Pattern

4.3.1 Pattern overview

Microkernel pattern is used for systems that should be adaptable and that should change based on the requirements (Buschmann et al, 1996). It encompasses a basic minimal core that provides the functionality to connect the different components and to provide the communication facilities between the system components. The services implemented by the microkernel are referred to as *mechanisms* and the extra functionality that is constructed on top of these mechanisms is called the system *policy*. The main problem addressed by the microkernel pattern is the need for adaptation based on evolving functional requirements and changing technologies.

4.3.2 Pattern solution

The microkernel pattern encapsulates the core service of the system into a microkernel component that has the following functionality:

- Facilitates communication between the different components of the system
- Responsible for management of system wide resources
- Provides the interfaces that allows components to access its functionality

The other system components in the microkernel pattern are (Figure 4-1):

1. Internal servers - these are the components that extend the functionality of the system. They implement extra functionality that the system needs and thus are the basic building blocks in terms of providing extensibility to the system. The internal servers are only accessed by the microkernel via service requests.

2. External servers - also known as personalities are components that implement a view of the underlying application domain. The clients communicate with the external servers, which in turn interpret the request, execute the request, and communicate the responses back to the clients.
3. Adapters - provide a communication layer between the clients and the associated external servers. Adapters protect the clients from the specific implementation details of the external servers.
4. Clients - these are the entities that require the services implemented by the system. Each client is associated with exactly one external server, and they execute the services by calling the interfaces on the external server.

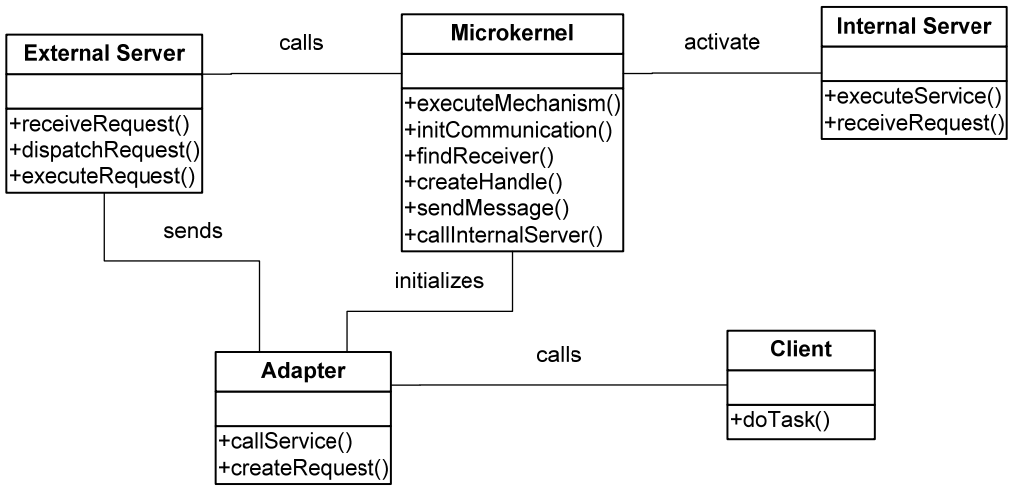

FIGURE 4-1 MICROKERNEL PATTERN

4.3.3 Pattern Discussion

The main benefit provided by the microkernel pattern is the enhanced ability to extend the system functionality and the ease of maintainability of the system. The drawbacks associated with the pattern are related to increased complexity of implementing extensibility measures in the pattern.

4.4 Internet Protocol Data Record (IPDR)

4.4.1 IPDR Overview

IPDR is an emerging standard that is developed by an open consortium that collaborates to facilitate the exchange of network usage and control data between the network elements and the Business Support Systems (BSSs). The main aim of this specification is to enable cost-effective usage management and data exchange to harness the implementation of next generation network services (IPDR, 2005). The specification offers a usage record that is open and that encapsulates the metrics and the parameters for any network service transaction. It also provides an extension mechanism that facilitates the exchange of optional service metrics and parameters. The standard specifies the encoding and transport protocols that can used in the exchange of the data records.

4.4.2 IPDR reference model

The IPDR reference model is based on the Telecommunication Management Forum's (TMF) Enhanced Telecommunication Operation Map (eTOM) for the purpose of defining the functional roles of different network elements and the interfaces between the elements and the BSSs. The eTOM model is used because it is the industry-accepted and widely implemented model of the telecommunication operations business processes. The IPDR model uses a layered architecture to define the different modules in the model (Figure 4-2).

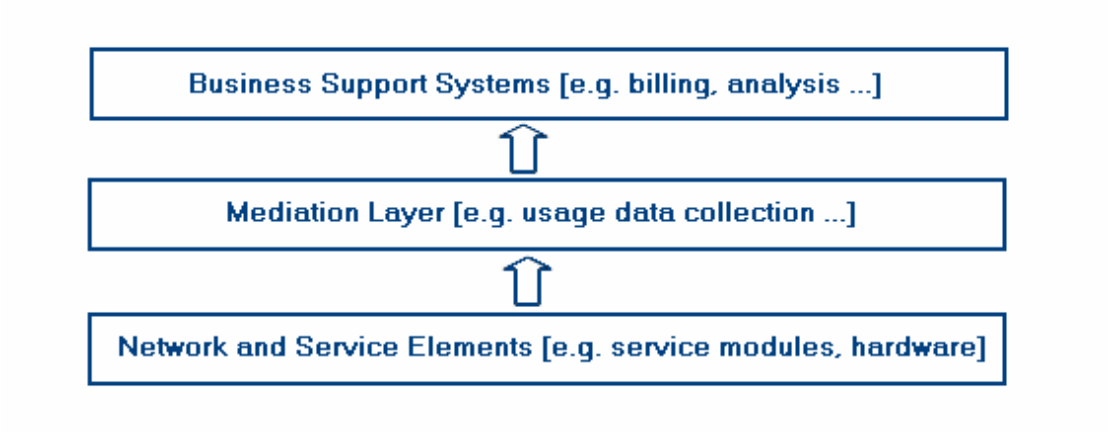

FIGURE 4-2 IPDR HIGH LEVEL MODEL

- Network and Service Elements layer - is made up of all the service modules and elements that essentially provide the network service to the clients. These are the elements that implement the basic functionality that makes up a service. This could include authentication services, web servers, file servers, and bandwidth management systems.
- Mediation layer - sits between the Network Elements (NE) layer and the BSS layer to provide an interface to the BSS and to the NE layer. The Mediation layer also serves the purpose of usage data collection; it collects the usage information from the NE layer, encodes the information and then passes it on to the BSS layer.
- Business Support Systems layer - supports the business operations that are relevant for the service provider. Some of the business operations that could be implemented in the BSS layer include, billing for the network usage and data mining operations based on the network usage.

The IPDR specifies different nodes in each of its layers and also defines the interfaces and the different interaction scenarios between those nodes. Figure 4-3 depicts the basic interaction between the nodes in the system.

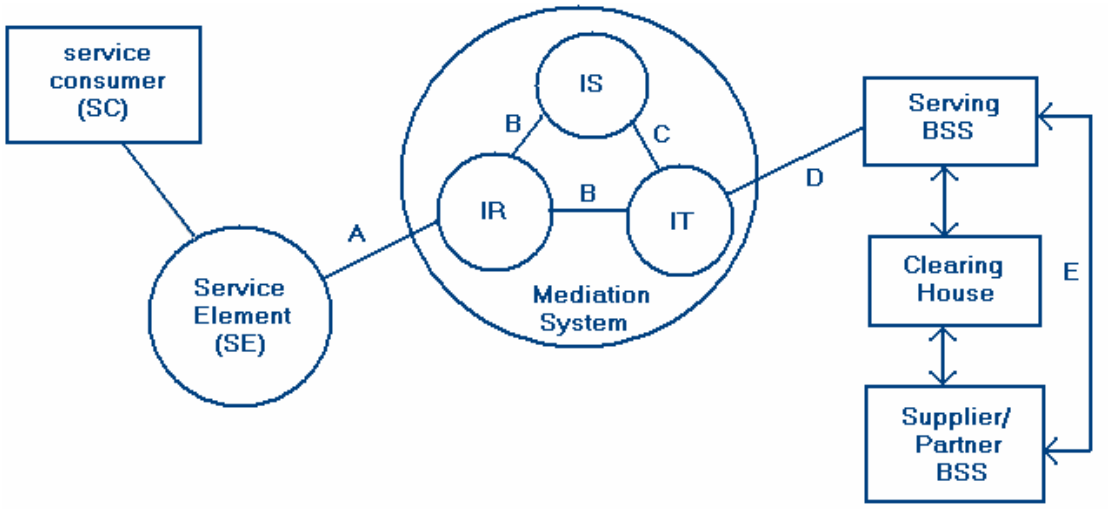

FIGURE 4-3 IPDR REFERENCE MODEL (IPDR, 2005)

- Service Consumer (SC) - is the entity that requests services from the SE. This is usually a user's device.

- Service Element (SE) - this is basic building block of the functionality of the system. It is usually a set of software modules that work together to provide a service or equipment that implements a certain required functionality (e.g., bandwidth management).
- Mediation System (MS) - this is the system that interfaces with SE on the one end and the BSS on the other hand. It comprises elements that collaborate to implement the mediation functionality:
    - IPDR Recorder (IR) – performs two basic functions. It collects the usage information from the SE, which may involve mediating proprietary protocols, and produces well-formatted IPDR documents from the collected information
    - IPDR Store (IS) – provides the necessary persistence for the IPDR documents by storing them in a format that can later be retrieved and retransmitted to other BSS if necessary.
    - IPDR Transmitter (IT) – delivers the IPDR documents to the BSSs. This element packages the IPDR from the IR and, if necessary, organises them according to service groups and then transmits them using the transport protocols that are implemented.
- BSS - this is the entity in the system that implements the business functions associated with the processes defined by the service provider. The serving BSS is the one that is associated with the system i.e. the local BSS and the Supplier/Partner BSS is that of the associated partners in a case of a roaming transaction. The clearinghouse may be necessary in the settlement process between the different service provider's BSS.

The IPDR reference model defines interfaces that facilitate communication between different model components. These interfaces are indicated in (Figure 4-3) and include:

- Interface A - this is the interface between the SE and the MS. This interface delivers usage information from the SE to the IR. The IPDR does not constrain the format, the naming conventions and the transport protocol to be used for this interface. This interface could implement proprietary protocols and conventions to communicate the usage data from SE.

- Interface B - interfaces the IR to the IS and the IT. It does not specify the transfer protocol to be implemented at the interface.
- Interface C - provides the connection between the IS and the IT.
- Interface D - delivers the IPDR documents from the MS (via IT) to the BSS. This interface defines the transfer to be used for the communication between the relevant components.
- Interface E - provides the connection from one BSS to another BSS, e.g. a supplier's BSS or a partner's BSS. This interface plays a crucial role in facilitating the implementation and operation of a settlement system.

4.4.3 IPDR information format

The exchange of the usage information between the different elements in the IPDR model is facilitated by a common information format that encapsulates the usage metrics of diverse network service. The IPDR record comprises 5 attributes that cover the different usage metrics of IP based network services:

- Who - this specifies the entity that is responsible for the usage of the network service. A user ID normally suffices for this attribute.
- When - this attribute specifies associated time that the network services usage took place (i.e., the end time or the event time).
- What - details the services that were utilised and the associated usage metrics and quantities (e.g., bytes, time duration, flows, hits, transactions). This attribute can also contain the QoS measures, state transition information and event codes.
- Where - this attribute provides traceability for the network service usage. It details the context within which the usage took place.
- Why - this normally specifies event trigger types (e.g., the reason why the NE is reporting a data element).

The IPDR record is based on XML, which allows it to leverage on the extensibility features of XML to include the usage attributes that are specific to a particular service that is implemented.

4.4.4 IPDR transport protocol

The IPDR specification specifies transport protocols that have to be used to transport service specific documents between the different entities in the reference model. This could be the transportation of information on any of the interfaces that are defined in the reference model. The main models of delivery that are supported by these transport protocols are:

- IT Push, where the IT is responsible for delivering the IPDR documents to the associated BSS
- BSS pull in which the BSS requests the IPDR documents from the IT
- Demand Poll, where the IT notifies the BSS of the availability of the IPDR documents and then the BSS pulls them as needed.

The IPDR specification currently defines File Transfer Protocol (FTP) and an IPDR streaming protocol (IPDR, 2005) as the main transport protocols to use with the implementation of IPDR.

4.5 Xobogel High-Level Model

The Xobogel architectural model (Figure 4-4) encompasses elements from the IPDR specification and the microkernel pattern to define a model that meets the requirements identified as necessary for small scale, easily deployable and extensible management system for PALs.

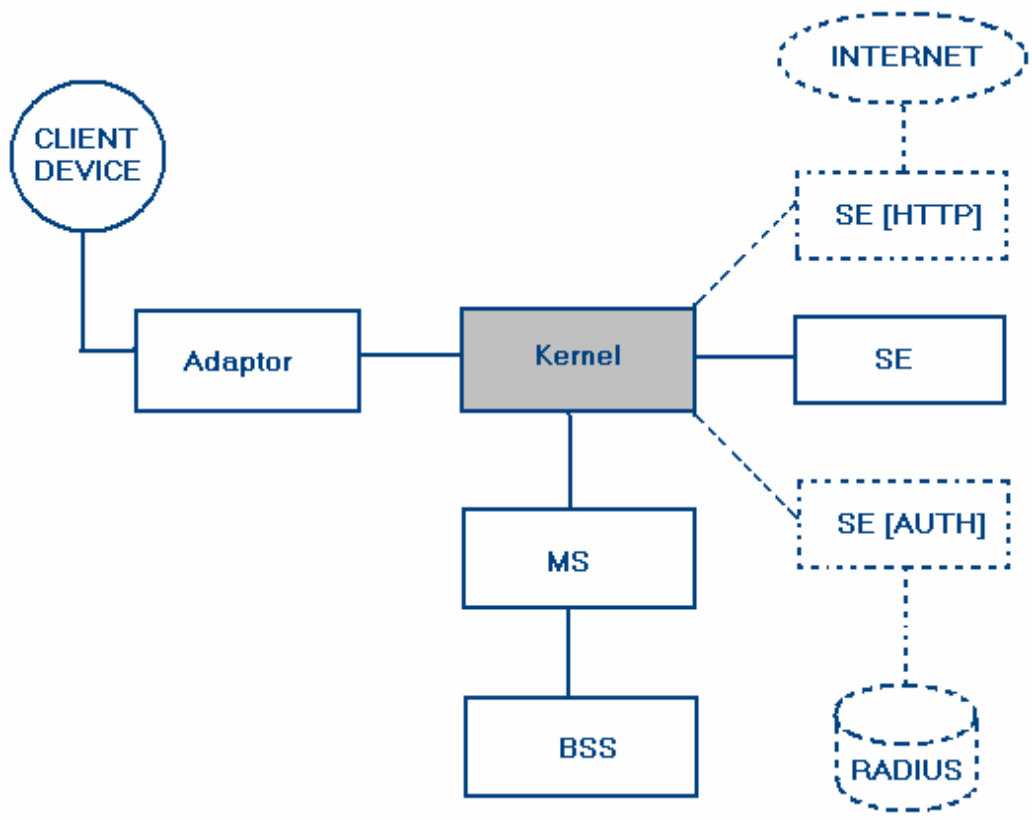

FIGURE 4-4 XOBOGEL HIGH LEVEL MODEL

This model comprises different components that collaborate to provide adequate services for the implementation of a PAL management system.

- Kernel - is the central core component of the model that implements minimal functionality to facilitate communication between the different components of the model.
- Service Element (SE) - provides the functionality that makes up a network service. The model highlights two examples of services that can be implemented in the system; the Authentication Service, which could be implemented via a RADIUS server, and the HTTP service that allows clients devices to browse the Internet. The service elements plug into the core Kernel. Extending the functionality of the system would essentially entail developing a new SE module and plugging it into the Kernel.
- Mediation System (MS) - implements the functionality needed to allow the BSS to operate of the information from the SE. It encapsulates the functions of

the IPDR Recorder, IPDR Store and the IPDR Transmitter. It collects the raw usage information from the SEs and formats it for usage by the BSS.

- Business Support System (BSS) - implements the business level operations that are necessary for the provision of the network service.
- Adaptor - provides a connection between the clients and the system functionality. It provides an interface into the system via which the clients can request services and receive results.

4.6 Xobogel Service Elements

The Xobogel framework implements a plug-in architecture that allows the different service elements to be plugged into the kernel. Each service element provides the functionality necessary for the provisioning of the service. For example, an authentication SE would provide the necessary mechanisms for authenticating users on the network (third-party authentication, RADIUS authentication or global servers' authentication), and a HTTP SE would provide network access to the Internet. Once a service element module has been loaded, the client devices are then able to request the service via an adaptor (Figure 4-5) that provides the communication channel to the service element.

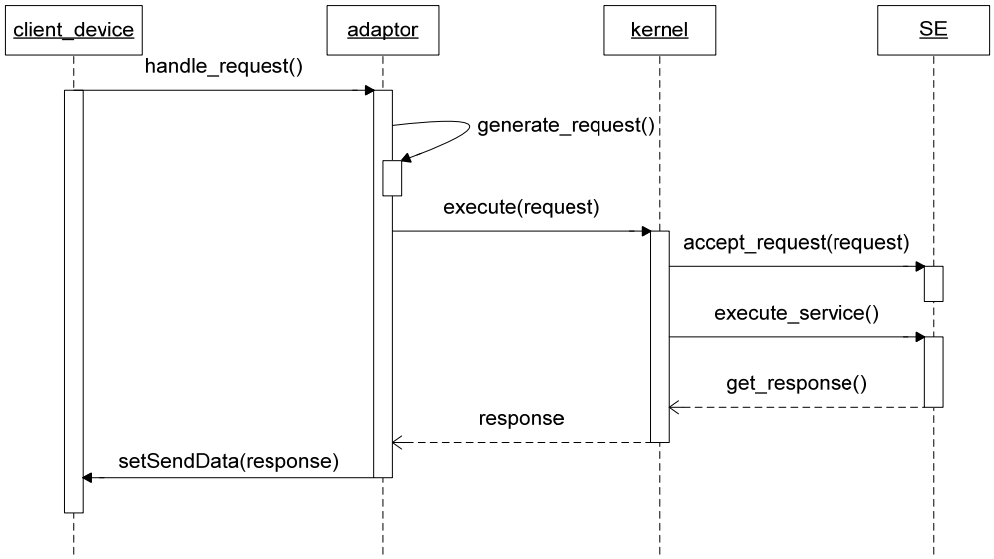

FIGURE 4-5 SERVICE ELEMENT USAGE SEQUENCE DIAGRAM

4.7 SE Extension Mechanism

The Xobogel framework defines, through an interface between the Kernel and SEs, a set of common operations that must be implemented by each SE. These set of operations provide the bare essentials for effective execution of SE modules. The following is a list of some of the operations, with examples from the HTTP SE, which are defined in the interface:

- `accept_request` – this is the method that accepts requests for a service. For the HTTP SE this method would accept formatted HTTP requests (Figure 4-5).
- `execute_service` – this method implements the functionality for the execution of the service. In the case of the HTTP SE, this would involve fetching the resource associated with requested URL and passing the relevant HTTP headers to the server (Figure 4-5).
- `get_response` – once the service request has been executed, the response is returned via this method. This equates to actual HTTP response in the case of the HTTP SE (Figure 4-5).
- `description` – provides a textual description of the SE for the purposes of usability and documentation.
- `get_schema_definition` – this method gets the XML schema definition for the usage metrics associated with the SE. This would return an XSD file (Listing 4-1) associated with the HTTP SE usage metrics.
- `get_usage_data` – this method returns the usage data for the SE. For the HTTP SE, it would return the actual network usage data, e.g. the IP addresses of the devices that made requests, the time the requests were made, and the bandwidth usage for each session.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
     targetNamespace = "http://www.ipdr.org/namespaces/ipd
     xmlns:ipdr = "http://www.ipdr.org/namespaces/ipdr"
     version = "3.0-A.0"
     elementFormDefault = "qualified"
     attributeFormDefault = "unqualified">
    <include schemaLocation = "IPDRDoc3.0.xsd"/>
    <element name = "clientIP" type="string"/>
    <element name = "startTime" type = "dateTime"/>
    <element name = "endTime" type = "dateTime"/>
    <element name = "usageUnit">
        <simpleType>
           <restriction base="string">
              <enumeration value="timeSecs"/>
              <enumeration value="timeMins"/>
              <enumeration value="dataBytes"/>
              <enumeration value="dataKbs"/>
           </restriction>
        </simpleType>
    </element>
    <element name = "quantity" type="int"/>
    <complexType name = "X_HTTP">
        <complexContent>
            <extension base = "ipdr:IPDRType">
                <sequence>
                    <element ref = "ipdr:clientIP"/>
                    <element ref = "ipdr:startTime"/>
                    <element ref = "ipdr:endTime"/>
                    <element ref = "ipdr:usageUnit"/>
                    <element ref = "ipdr:quantity"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

LISTING 4-1 A SAMPLE XSD FILE

The SE modules can define further operations that provide the functionality for the implementation of the network service. Further functionality that needs to be provided by the framework is for the purpose of meeting the Authentication Authorization and Accounting network management requirements, in particular the need to account for network usage.

4.8 Xobogel Network Usage Accounting

In order to facilitate network usage accounting for the PALs, each SE module defines the metrics associated with its usage. This is done via an XML schema definition file which is used by the mediation system to format the usage data into IPDR Document format.

For example, a HTTP service element module could define simple usage metrics shown in (Table 4-1). The data type and the category associated with each usage attribute should be determined and documented.

TABLE 4-1 USAGE METRICS FOR A NETWORK SERVICE

Category	**Usage Metric**	**Data type**
Who	*ClientIP*	String
When	*StartTime*	Date/Time
When	*EndTime*	Date/Time
What	*UsageUnit*	String
What	*Quantity*	Integer

Subsequent to the identification of the usage attributes associated with a SE module, an XML schema definition file that implements the IPDR specification is developed (Listing 4-1). This XSD file is used by the mediation system to determine how the usage data should be formatted.

```
<?xml version="1.0" ?>
- <IPDRDoc xmlns="http://www.ipdr.org./namespaces/ipdr" xmlns:xsi="htt
    xsi:schemaLocation="http://www.ipdr.org./namespaces/ipdr x_http.x
    0a9bc0a80064" creationTime="2005-01-22T08:38:41.392Z" IPDRRecord
  - <IPDR xsi:type="X_HTTP">
      <IPDRCreationTime>2005-01-22T08:38:41.392Z</IPDRCreationTime>
      <seqNum>1</seqNum>
      <clientIP>192.168.0.101</clientIP>
      <startTime>-1718481985</startTime>
      <endTime>-1718481438</endTime>
      <usageUnit>dataBytes</usageUnit>
      <quantity>2664</quantity>
    </IPDR>
  - <IPDR xsi:type="X_HTTP">
      <IPDRCreationTime>2005-01-22T08:38:41.392Z</IPDRCreationTime>
      <seqNum>2</seqNum>
      <clientIP>192.168.0.101</clientIP>
      <startTime>-1718481188</startTime>
      <endTime>-1718481157</endTime>
      <usageUnit>dataBytes</usageUnit>
      <quantity>3323</quantity>
    </IPDR>
```

LISTING 4-2 FORMATTED USAGE INFORMATION

Once the data has been formatted (Listing 4-2), it is then accessible to the BSS or any other business unit for further processing. This allows every SE to be self describing and self contained (i.e., each SE encapsulates the functionality to execute the required service and also the information on how to account for its usage). The ability of SEs to be self describing also facilitates the exchange of usage information with business partners (e.g., roaming partners, clearing houses).

4.9 Benefits of the Architecture

The Xobogel architecture combines the flexibility and adaptability of the microkernel pattern with the portability of the IPDR document format. The framework provides the following benefits which are verified by the implementation in chapter 5 and the experimentation in chapter 6 (Thinyane et al, 2005):

- Reliability - due to the separation of the SEs from the rest of the system, a more robust architecture is achieved which isolates service related problems from crippling the whole system i.e. a problem in the HTTP service module does not bring down the Authentication service module.

- Portability - the architecture separates the service-specific implementation factors from the high level system management considerations. The kernel is separated from the SEs by a level of abstraction that defines an interface that should be implemented by the SE.
- Modularity - the architecture comprises components that are modular and that encapsulate elements of the system functionality.
- Flexibility - the decoupled architecture allows for flexibility in terms of changing and modifying the components of the system without having to alter the whole system. SE modules can be added and launched on the fly without having to bring the whole system down. If a WISP does not need authentication service at the PAL they simply remove the authentication SE from the system.
- Integration into WISP business systems is easily achieved. This is due to the fact that the architecture provides the network usage data in an IPDR Doc format which allows WISPs to use it as fits their business model.
- Ease of implementation - the framework implements an established and well-understood architecture, the microkernel pattern. The functionality related to the IPDR specification is also easy to implement due to the free availability of open IPDR libraries. The framework's plug-in architecture allows for the system to be implemented using already developed components e.g. service elements, billing modules.

4.10 Summary

An architectural framework named Xobogel is derived to meet the requirements previously identified in Chapter 3 for small PALs. The basic building blocks of the framework are the microkernel pattern and the IPDR specification which provide system extensibility and an extensible usage accounting framework respectively. The framework provides for the extensibility of both the associated service element modules and the billing modules implemented at a PAL. The service elements are implemented in a self-contained module that defines an XML Schema Definition (XSD) file that is used to format the usage data into IPDRDoc format. Some of the benefits realised through the architecture include: reliability, portability, modularity and ease of integration with the BSSs.

5 Chapter 5: The Xobogel based PAL solution

The previous chapter provided an architectural framework for the implementation of a system that would address the requirements associated with small PAL management. This chapter provides a description of the implementation of the system named SEHS (Small Extensible Hotspot System). SEHS implements the Xobogel framework and has been developed as a proof-of-concept prototype to validate the applicability and the effectiveness of the Xobogel framework in meeting the previously identified requirements (Chapter 3).

5.1 System Hardware components

The SEHS system is simple and comprises an AP for the provisioning of the wireless interface to the system, the system PC that hosts the SEHS application and the associated databases (Figure 5-1).

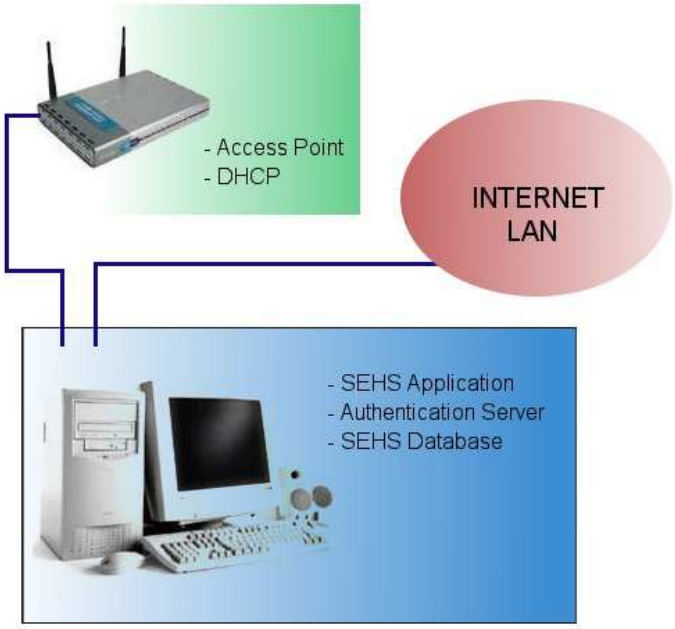

FIGURE 5-1 SEHS SYSTEM OVERVIEW

The system PC has two NICs, one for connection to the AP and the other for connection to the LAN/Internet. The specification of these hardware components are as follows:

5.1.1 D-Link Access Point

A D-Link DWL-2000 access point (802.11g/b, 2.4GHz with OFDM) is used in the implementation of SEHS. The DWL-2000AP provides the functionality that is needed to facilitate connectivity between the mobile devices and the LAN. Its also provides a number of management features that are essential in the provisioning of network services by small PALs such as:

- Web configuration - The configuration of the DWL-2000AP can be done via a web browser. This feature is crucial in the light of the fact that the level of technical expertise at the different PALs may be limited. A web interface for the AP configuration therefore increases the usability of the AP, in terms of ease of use.

- Authentication Options - The DWL-2000AP offers a number of different authentication options that can be implemented on the wireless interface.
    - Open system - all the devices that request connection to the AP are accepted without any authentication measures being implemented.
    - Shared key: this option allows devices with the right keys to connect to the AP.
    - WPA - WiFi Protected Access is a WLAN standard that was introduced to replace WEP and to address some of the security concerns of the WEP standard. It authorizes and identifies used based on a secret key that is changed regularly. It works in conjunction with a RADIUS server and employs the TKIP for key management.
    - WPA-PSK - is a WiFi Protected Access mechanism that uses Pre-Shared Keys.
- WEP encryption - The DWL-2000AP provides WEP encryption functionality (see section 2.5.2.1).

- DHCP - In order to facilitate nomadic usage of PALs (see section 2.5.1), the DWL-2000AP provides a DHCP functionality to assign IP addresses to the mobile devices that connect to the PAL.
- MAC filtering - An added security functionality that can be implemented on the DWL-2000AP is the MAC filtering function which allows only the specified (in the MAC filter table) computers to connect to the AP. This functionality introduces a limitation onto the PAL in terms of requiring the users to first have their MAC recorded before using the network service.
- 802.1x - Further security features provided on the DWL-2000AP include the 802.1x port authentication functionality (see section 2.5.2.2).
- AP status reporting function - A number of reporting logs are generated for the usage of the DWL-2000AP and these can be view via the web interface. Some of these logs include the DHCP table and connection mobile devices log.

5.1.2 Computer

The SEHS system is implemented on an Intel P4 3.0 GHz processor, with 516 MB RAM and 40 GB disk space. The two NICs used on the PC are an Intel PRO/100 VE card and the Realtek RTL8139 Family PCI Fast Ethernet card. The PC is running the Window XP Professional operating system.

5.1.3 Mobile Devices

One of the mobile devices that was used in this project is the Fujitsu Pocket LOOX PDA. The PDA is running on MS Windows Mobile 2003 operating system on an Intel XScale PXA255 400 MHz processor. The wireless connectivity provided on the PDA is via Bluetooth, IEEE 802.11 and IrDA. The PDA has 64 MB RAM and a TFT 240 * 320 3.5” display.

The other mobile device utilized in the project is the HP Onmibook XE_3. The laptop runs the Windows XP Professional Operating System on an Intel P4 1066 MHz processor, with 256 MB RAM and 20 GB disk space.

5.1.4 The system as a unit

The central component in the hardware infrastructure is the PC (henceforth called SEHS PC) running the SEHS system. It is the one that intercepts communication from the client devices and subsequently handles all the requests for network resources on behalf of the clients. To facilitate this interception behaviour, the SEHS PC has two NICs installed, one connecting to the LAN and the other connecting to the AP. Configuring the SEHS PC to function properly in MS Windows simply involves altering the routing table (see Appendix A: SEHS PC configuration).

The AP is configured to provide DHCP functionality to the requesting devices. Therefore all the mobile devices that request connection to the AP are assigned an IP address within a specified range (192.168.0.100 – 192.168.0.199), which should be sufficient to cater for IP addressing needs of a small PAL. The second NIC on the SEHS PC is also assigned an IP address by the AP so that it's in the same network subnet as the rest of the requesting client devices.

5.2 Implementation variables

Since the architectural framework is designed to function as a guideline for the implementation of the system, it can be implemented using any language and on any platform that provides the basic networking functionality needed for PAL services. This particular implementation of the system is based on the JAVA language and is being implemented for the Windows OS.

5.2.1 Use of JAVA

SEHS is implemented using the JAVA 1.5 SDK. The decision to use JAVA is influenced by the following advantages that this language provides for meeting the requirements for small PALs:

- Low cost - The JAVA SDK is free of charge and this reduces the Total Cost of Ownership (TCO) associated with implementing SEHS.
- Platform independence - The main advantage of JAVA is its portability and this is an essential feature as it would allow SEHS to be implemented on different operating systems depending on the WISPs preferences.

- Class libraries - JAVA provides a growing collection of class libraries that are applicable for a wide range of application domains.
- JAVA Virtual Machine (JVM) - Shields the rest of the computing environment from problems that arise in the JAVA code (i.e., a problem in a JAVA application does not bring down the whole system down rather just the JVM).
- JAVA is a dynamic language - Which means that it provides features that allow it to adapt to an evolving environment and it also for runtime loading of JAVA classes.
- Object orientation - The fact that JAVA is object oriented facilitates the development of modular, flexible and reusable code.
- JAVA is easy to learn - JAVA was designed to be easy to use and hence easy to write, compile, debug. The simplicity in the language is provided by features that include (IBM, 2004):
    - The use of an interface structure to eliminate the complexities of multiple inheritances.
    - JAVA provides an automatic memory allocation which removes the need on the part of the programmer to do memory allocation in code.
    - The garbage collection feature in JAVA eliminates the need for programmers to collect garbage.
    - The number of language constructs is relatively smaller than in other languages.

The disadvantage of using JAVA, though negligible in this implementation, is the fact that it is an interpreted system and therefore relatively slower than compiled languages. Program execution in JAVA involves the byte code being converted by the JVM into native instructions which correlate to the platform's instruction set. This conversion cycle consumes some processor time (Choudhari P., 2001).

5.2.2 Implementing on the Windows platform

Windows is the OS of choice in this particular implementation because it is widely used among a large portion of small businesses, including those that were profiled in Chapter 3. Thus it would make it easier to adopt SEHS on their current OS.

The costs associated with acquiring the MS Windows licence is one of the main prohibitive factors that would lead to consideration of alternative platforms. The total cost of ownership of MS Windows is also increased by the costs associated with dealing with viruses, spyware, and malware.

In a study undertaken to compare the total cost of ownership of Ms Windows and Linux, it was found that the total cost of ownership is higher for Windows than it is for Linux. This was taking into consideration the factors related to installation of the operating system, usability of the system, maintenance of the system, system upgrades, and purchase of application software (Cybersource, 2002).

5.3 SEHS UML

UML is the underlying object modelling technique that has been used for the design of SEHS. UML provides various tools that can be used to describe the static structure of a system, the relationships between the data objects and the dynamic behaviours implemented in a system. The sections below provide a description of the system using the UML notation.

5.3.1 Object Diagram

The SEHS object diagram (Figure 5-2) indicates the various objects that are implemented in the system to provide the overall functionality that is encapsulated in the system. These objects are closely related to the microkernel objects in terms of the functionality that they provide. The object diagram (Figure 5-2) also provides an overview of the behaviour that is provided by each of the objects in the system.

- `NioSocket` - is a class that provides the non-blocking socket framework on the JAVA platform. This class is based on the JAVA NIO API which provides, via the selector object, the ability to handle hundreds of simultaneous I/O requests. The JAVA NIO API eliminates the high thread overhead associated with the 1:1 thread to client ratio of the pre JAVA 1.4 network applications (Shirazi J., 2005).
- `x_adaptor` - provides a communication layer between the clients and the engine. The `x_adaptor` is responsible for reading and writing to the client devices. It overrides the methods in the `NioSocket` to provide system-

specific implementation of network operations (i.e. protocol related socket operations).

- `x_engine` - this is the core object in the system and it provides the functionality for facilitating communication between the different system objects, management of system wide resources and providing an interface for objects to execute the functionality provided by the service elements.
- `x_main` - is responsible for launching the system.
- `x_me` - this is the mediation system that handles all the usage accounting related functions of the system. It provides the functionality to read the usage data from the service elements and to format it to the IPDR Doc format. It provides persistence to the IPDR Doc and also provides the functionality to forward the IDPR Docs to the BSS
- `x_bss` - provides the business functionality related to the provisioning of the network service (e.g. billing and data mining).
- `x_serviceElement` - this interface represents the basic unit of network service functionality implemented in the system. Most of the network service specific functionality provided in the system is encapsulated in the `x_serviceElement` object.
- `x_database` - this is the class that provides the functionality needed to perform operations on the system database.
- `x_billModule` - is the module that is associated with determining the cost of the usage of the network service based on the usage metrics from the service element modules.
- `x_request` - this is the object that represents client's request for a network service.
- `x_response` - represents responses that are sent back to the requesting clients.

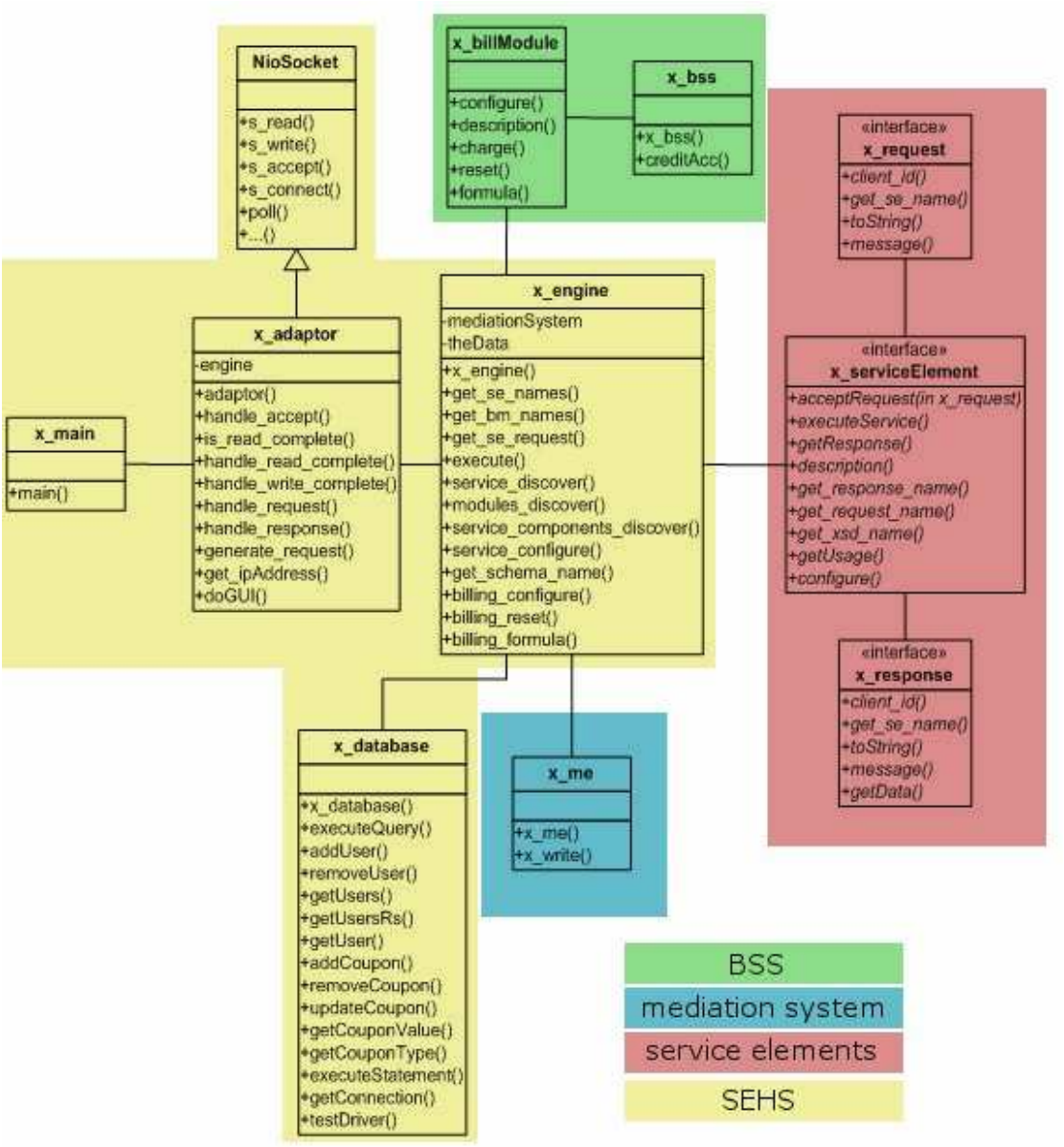

FIGURE 5-2 SEHS OBJECT DIAGRAM

5.3.2 Use Cases

These Use Cases detail the steps involved in the typical usage of a system based on the Xobogel framework and also highlights the interactions between the associated components of the system.

5.3.2.1 Use Case: Service Request

The basic usage flow (Figure 5-3) for a client device requesting a service on the network is as follows:

- The ***Client Device*** associates with an ***Adaptor*** which then handles the rest of the communication between the ***Client Device*** and the system ***Engine,*** also known as `x_engine` (Figure 5-3 A)
- The ***Adaptor*** accepts a request from the ***Client Device*** (Figure 5-3 A).
- The ***Adaptor*** generates a request and sends it to the ***Engine*** (Figure 5-3 B).
- The ***Engine*** executes the request on the associated ***Service Element*** (Figure 5-3 C).
- The service response is passed from the ***Service Element*** to the ***Client Device*** via the ***Adaptor*** (Figure 5-3 C, B, A)

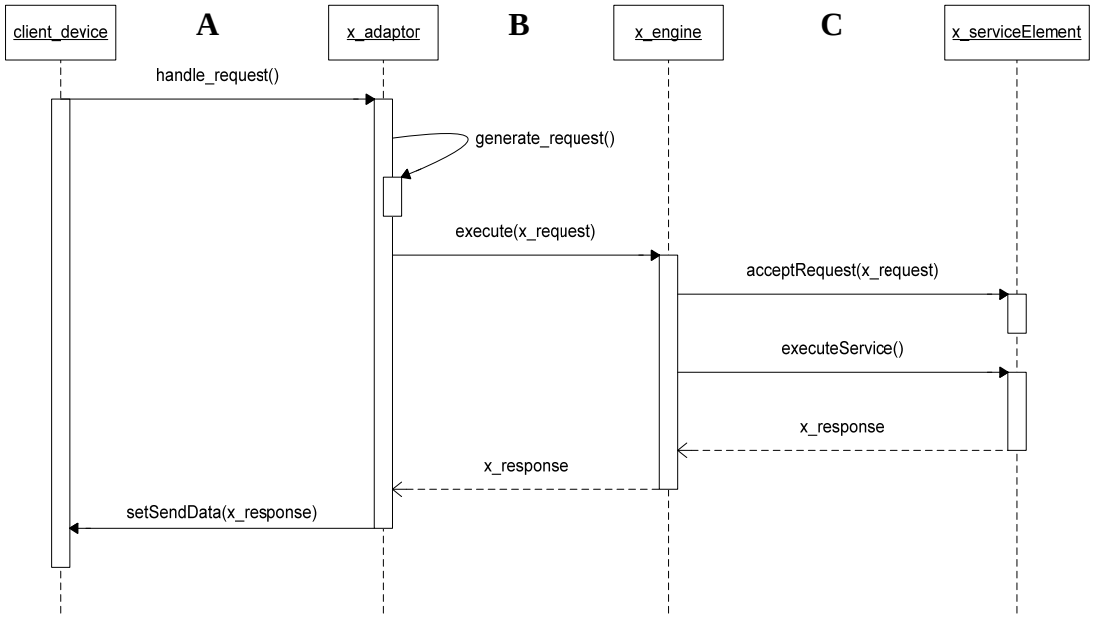

FIGURE 5-3 SERVICE REQUEST SEQUENCE DIAGRAM

5.3.2.2 Use Case: System Start-up

This use case highlights the processes that are undertaken when the system is launched (Figure 5-4):

- The ***Adaptor*** is initialised (Figure 5-4 A).
- The ***Adaptor*** initialises the ***Engine*** (Figure 5-4 B).
- The ***Engine*** determines the available ***Service Elements*** (Figure 5-4 C).

- The ***Engine*** generates an aggregation of all the ***Service Elements*** it has loaded and exposes these to the ***Adaptor*** (Figure 5-4 C).

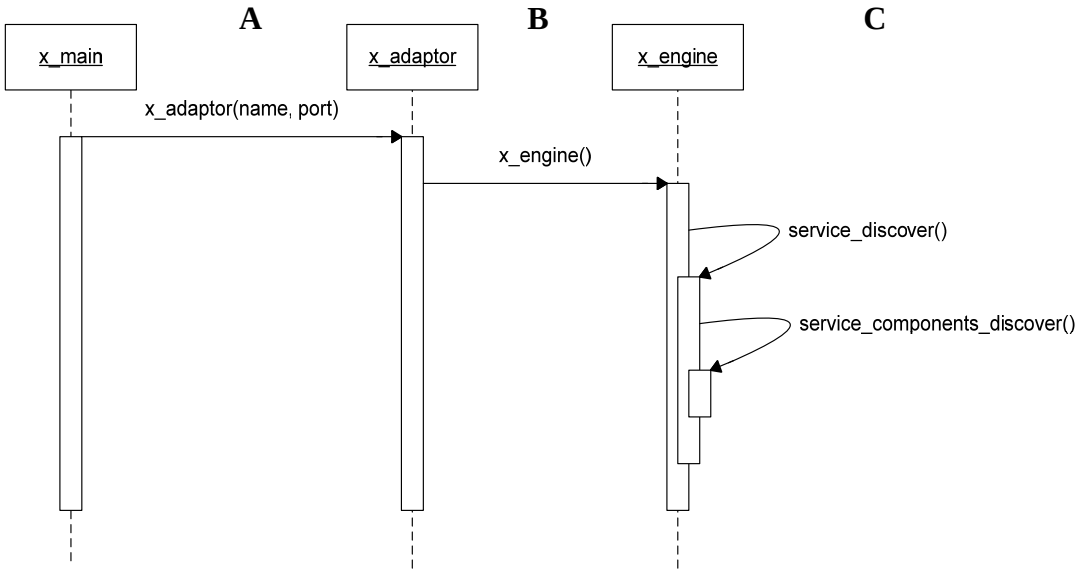

FIGURE 5-4 SYSTEM START-UP SEQUENCE DIAGRAM

5.3.2.3 Use Case: Usage accounting

Usage accounting scenario (Figure 5-5) encapsulates the functionality of collecting and generating usage records that are then formatted into a standard format that can be interpreted by the BSS layer modules.

- An ***Adaptor*** requests a service on behalf of a ***Client Device*** via the ***Engine*** (Figure 5-5 A).
- After executing the request, the ***Engine*** queries the ***Service Element*** for the usage data and then forwards this to the ***Mediation System*** (Figure 5-5 B, C)
- The ***Mediation System*** formats the usage information into the IPDR format and writes it to a persistent storage.

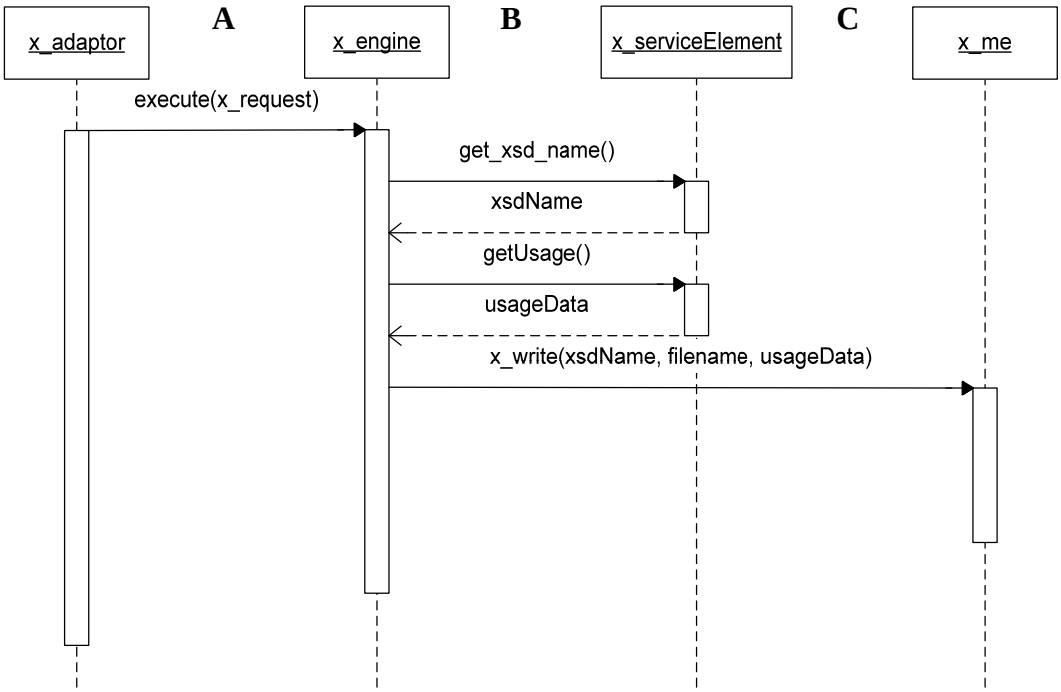

FIGURE 5-5 USAGE ACCOUNTING SEQUENCE DIAGRAM

5.3.2.4 Use Case: Billing

Billing scenario (Figure 5-6) shows the processes involved in billing for the network usage based on the usage data from the service element modules.

- An ***Adaptor*** requests a service on behalf of a ***Client Device*** via the ***Engine*** (Figure 5-6 A).
- The ***Engine*** determines if charging is enabled (Figure 5-6 B).
- If charging is enabled, the ***Engine*** forwards the usage data to the ***Billing Module*** (Figure 5-6 B, C).
- The ***Billing Module*** calculates the charge for the usage and forwards this information to the ***BSS*** (Figure 5-6 D).
- The ***BSS*** in turn updates the system database to reflect the billing information.

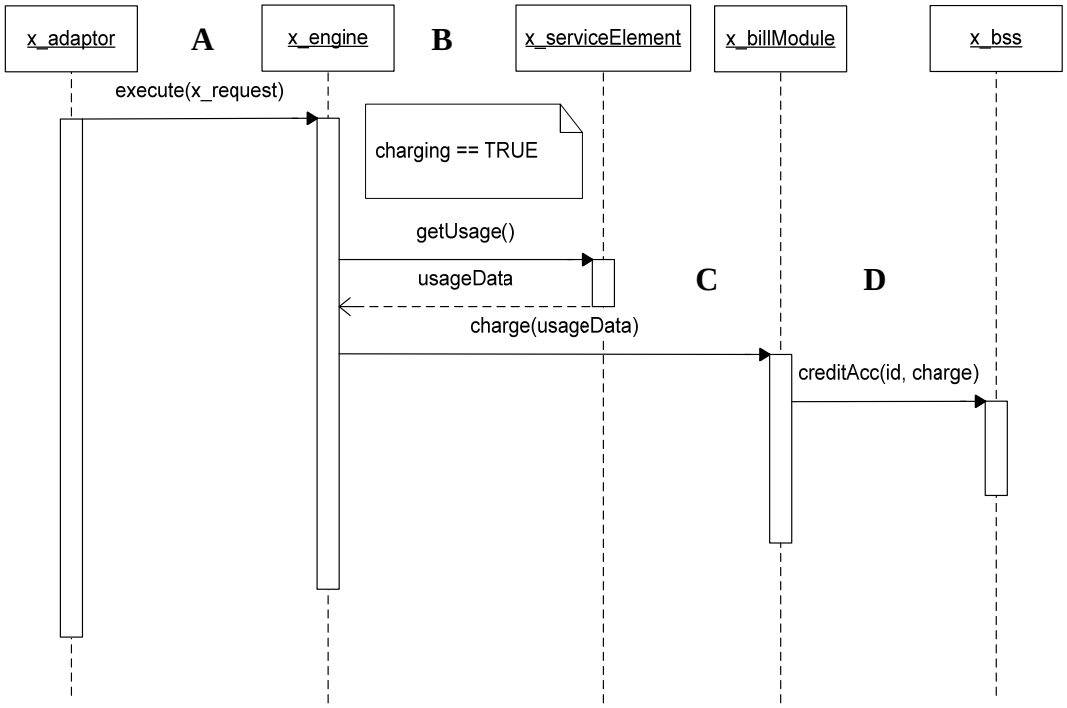

FIGURE 5-6 BILLING SEQUENCE DIAGRAM

5.4 Implementing Services

5.4.1 Xobogel services interface

`Xobogel` service modules must implement the `x_serviceElement` interface (Figure 5-7) defined in the Xobogel framework. The associated request and response objects must implement the `x_request` and `x_response` interfaces respectively. This facilitates a standard means of communication between the `x_engine` and the service elements.

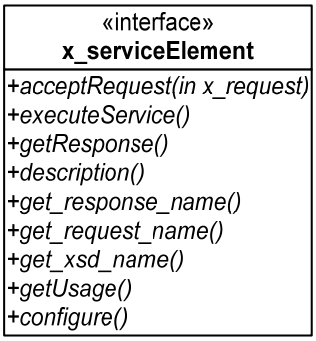

FIGURE 5-7 SERVICE ELEMENT INTERFACE

The service element interface (`x_serviceElement`) defines the following methods:

`acceptRequest()`

This is the method that receives the associated request object for the service element. The request object MUST implement the *x_request* interface in order to facilitate the communication with the other system components. The *x_request* interface (Figure 5-8) defines the following methods; *client_id()* which is an accessor method for the client device identification, *get_se_names()* returns the class name for the associated service element, and *message()* which returns the data encapsulated by the request object.

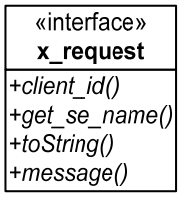

FIGURE 5-8 REQUEST INTERFACE

`executeService()`

This method executes the service based on the request object. This is the bulk of the functionality that is offered by the service element. This method returns a response object associated with the service element. The response object must implement the *x_response* interface which defines *client_id(),* an is an accessor method for the client device identification, *get_se_names()* which returns the class name for the associated service element, *message()* returns the data encapsulated by the response object as string and *getData()* returns the encapsulated data as a byte array.

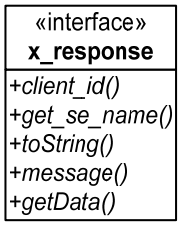

FIGURE 5-9 RESPONSE INTERFACE

`getResponse()`

Returns the results of the service execution encapsulated in the response object. This method offers an alternate means of accessing the response from the execution of the service.

`description()`

Each service element should be self describing (in a human readable form) and this method returns a definition of a service element.

`get_response_name()`

Returns the class name of the response object associated with the service element module. This method is used in the initialisation of the system components which need to be instantiated before execution begins.

`get_request_name()`

This method returns the class name of the request object associated with the service element and is used for instantiating system components i.e. service element modules components.

`get_xsd_name()`

This method returns the name of the XML schema definition file that specifies the usage fields associated with the usage records generated by the service element. The schema definition file is for the formatting of usage records by the mediation system. This allows for the seamless exchange of usage information across business units and between business partners based on the common service element schema definition. The service element schema is extended from the IPDR schema and defines the fields

that represent the usage data. Every service element should have a schema definition in order for the mediation system to format the IPDRDocs accordingly.

`getUsage()`

This method returns the usage data for the service element. The usage data should be in a format that the mediation system can interpret and hence format using the associated schema definition.

`configure()`

SEHS interface for the SE modules defines a configure method that must be implemented by the service elements that are used on the system. This method allows for the configuration of the SE modules via the GUI that is implemented in the SE module.

5.4.2 Example Service Element (`x_se_http`)

One of the top uses of PALs, as ascertained in section 3.2, is web browsing. The web browsing functionality in SEHS is provided via the `x_se_http` service element. The description of this service element module and its implementation is hereafter detailed. A description of the other service element modules implemented in SEHS is detailed in Appendix B: Example service elements.

5.4.2.1 `x_se_http` description

This service element module provides HTTP functionality to browse the Internet. It intercepts the client's HTTP requests, forwards them to a web server or a web proxy server, reads the response from the web server and forwards the response back to the client. It also provides the associated usage accounting functionality.

5.4.2.2 `x_se_http` usage scenario

The steps below detail the usage of the `x_se_http` service element module and its interaction with other system modules.

- A request (`x_req_http`) is received from the `x_engine`
- The service element module (`x_se_http`) determines the URL of the requested resource from the request and fetches the resource. It handles

both local resources and remote resources, in which case it provides web proxying functionality.

- A response (`x_res_http`) is constructed from the information fetched from the internet
- The response is send back to the request device

5.4.2.3 `x_se_http` object diagram

The data objects associated with the `x_se_http` service element module are depicted in Figure 5-10.

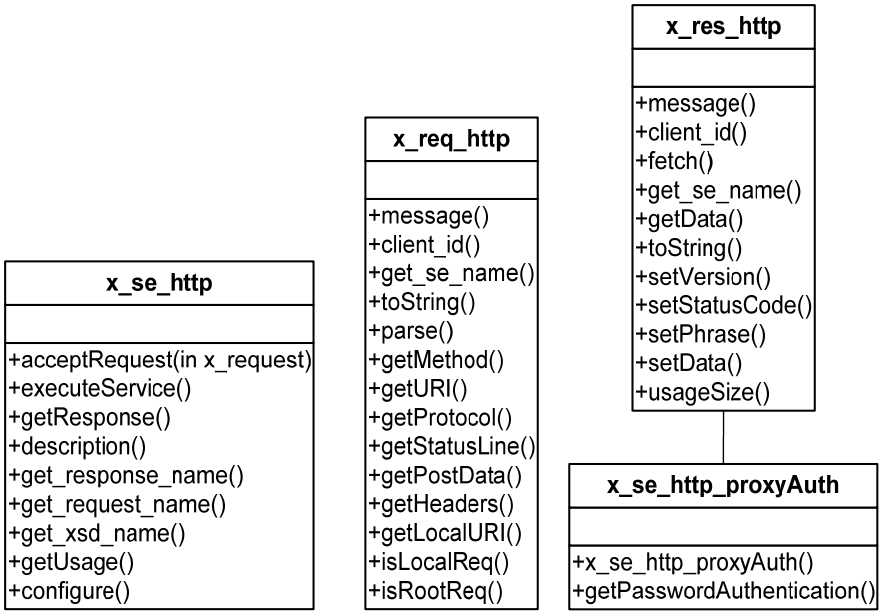

FIGURE 5-10 X_SE_HTTP DATA OBJECTS

Implementing the HTTP service is achieved by compiling the associated service element object (`x_se_http`), the request object (`x_req_http`) and the response object (`x_res_http`). The `x_res_http` object is associated with a `x_se_http_proxyAuth` object in order to allow the service to operate through a web proxy.

5.4.2.4 `x_se_http` usage accounting

Accounting for the `x_se_http` service element usage in SEHS is achieved via the mediation system which is implemented in the `x_me` data object. The mediation system takes as its input the schema definition file (Listing 5-1) of the usage metrics associated a service element module. The `x_se_http` service element usage metrics are defined in Table 5-1, categorised according to the IPDR attributes categories discussed in section 4.4.3.

TABLE 5-1 X_SE_HTTP USAGE METRICS

Category	**Usage attribute name**	**Data type**	**Presence**
Who	ClientIP	String	REQUIRED
When	StartTime	Date/Time	REQUIRED
When	EndTime	Date/Time	REQUIRED
What	UsageUnit	String	REQUIRED
What	Quantity	Integer	REQUIRED

```
<?xml version = "1.0" encoding = "UTF-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
     targetNamespace = "http://www.ipdr.org/namespaces/ipdr"
     xmlns:ipdr = "http://www.ipdr.org/namespaces/ipdr"
     version = "3.0-A.0"
     elementFormDefault = "qualified"
     attributeFormDefault = "unqualified">
    <include schemaLocation = "IPDRDoc3.0.xsd"/>
    <element name = "clientIP" type="string"/>
    <element name = "startTime" type = "dateTime"/>
    <element name = "endTime" type = "dateTime"/>
    <element name = "usageUnit">
        <simpleType>
           <restriction base="string">
              <enumeration value="timeSecs"/>
              <enumeration value="timeMins"/>
              <enumeration value="dataBytes"/>
              <enumeration value="dataKbs"/>
           </restriction>
        </simpleType>
    </element>
    <element name = "quantity" type="int"/>
    <complexType name = "X_HTTP">
        <complexContent>
            <extension base = "ipdr:IPDRType">
                <sequence>
                    <element ref = "ipdr:clientIP"/>
                    <element ref = "ipdr:startTime"/>
                    <element ref = "ipdr:endTime"/>
                    <element ref = "ipdr:usageUnit"/>
                    <element ref = "ipdr:quantity"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

LISTING 5-1 X_SE_HTTP SCHEMA DEFINITION FILE

5.4.2.5 `x_se_http` screenshots

The compiled service element modules are made available for use in SEHS by loading them into the system modules directory. The system automatically scans the modules directory for service element modules and loads them into the system for configuration. Figure 5-11 shows the `x_se_http` in the list of available service element modules. Once the `x_se_http` module is set as the current service element, all the clients requests are handled and executed via the new service element.

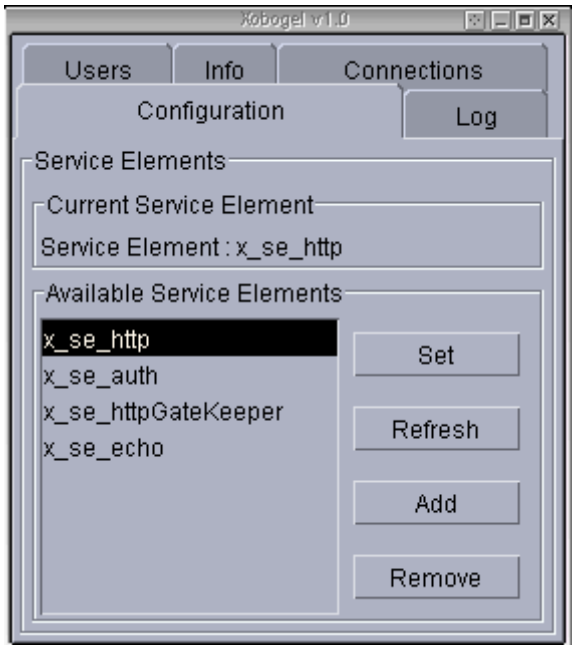

FIGURE 5-11 X_SE_HTTP SERVICE ELEMENT CONFIGURATION

Figure 5-12 shows the operation of the `x_se_http` module in the provisioning of a web browsing service to a PDA.

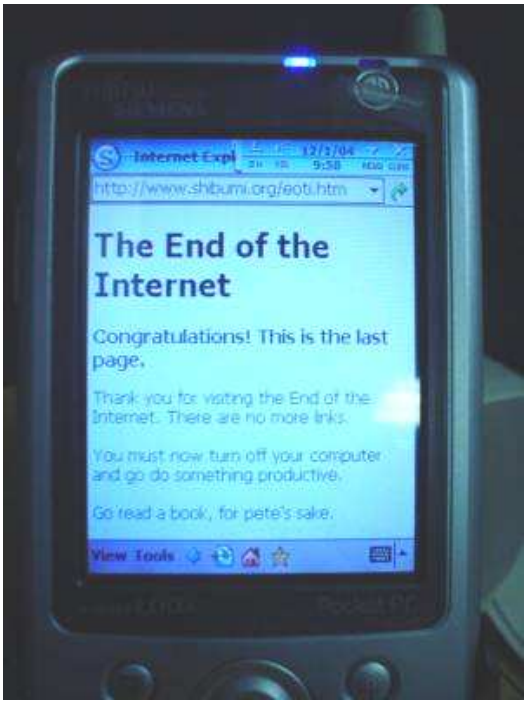

FIGURE 5-12 X_SE_HTTP SERVICE ELEMENT UTILIZATION

5.4.3 Implementing Extensibility in JAVA

The microkernel architecture which forms the basic building block of this system provides an inherent mechanism for extending the functionality of the system. The actual implementation of this in JAVA is achieved by using a feature of JAVA that allows runtime loading of classes. Dynamic loading is achieved by using the *forName()* method of the Class object. The method takes a name of the JAVA class that should be loaded by the JVM.

In SEHS, the SE modules are loaded in a similar fashion during the system start up. All the classes in the modules directory that implement the defined interfaces are loaded and made available to the system for execution. These classes are loaded into a HashTable to allow for an easy, indexed access to the class objects as they are needed by the system (Listing 5-2). The extensibility that is provided in this manner is for the service element modules and for the billing modules.

```
if(is_x_se || is_x_bm)
 {Object class_obj = (Class.forName(filename)).newInstance();

  if(is_x_se)
     {x_ses.put(filename, class_obj);
     }

  if(is_x_bm)
     {x_bms.put(filename, class_obj);
     }
 }
```

LISTING 5-2 SYSTEM MODULES DYNAMIC LOADING

5.5 SEHS Functionality

The system provides the functionality that is needed for the provisioning of PAL network service. The other functionality worth highlighting in this system follows:

5.5.1 Authentication

The authentication implemented in SEHS is built on the Open System Authentication (OSA) architecture of the 802.11 protocol. All devices are allowed to associate with

an AP and depending on the requirements of the executing service element; authentication is only performed on service requests from the clients. Authentication is not a PAL wide requirement as ascertained in the requirements for the PALs i.e. some PAL service may need client authentication others may not, therefore authentication is implemented via a service element module called `x_se_auth` (Appendix B: Example service elements) and this module is utilized by all the other service element modules that require authentication. In the current implementation of the system the modules that implement authentication functionality are the `x_se_httpGateKeeper` and the `x_se_httpCouponKeeper` SE modules. To facilitate the implementation of the authentication infrastructure, the following connection states are defined in these SE modules (Listing 5-3):

```
/** let the request execute on the server */
public final int X_LET_IN = 1;
/** reject the request and block it from executing */
public final int X_REJECT = 2;
/** authenticate the request on the server */
public final int X_AUTHENTICATE = 3;
/** invalid authentication credentials - reauthenticate */
public final int X_INVALID = 4;
/** successfully authenticated the client */
public final int X_AUTHENTICATED = 5;
/** successfully logged out of the system */
public final int X_ADIOS = 6;
```

LISTING 5-3 AUTHENTICATION STATUSES

- `X_LET_IN` - the client has already been authenticated on the system (i.e., let the request in).
- `X_REJECT` - the client is blocked from using the network service, therefore reject this client's request.
- `X_AUTHENTICATE` - redirect this request to the login page and execute the authentication procedure.
- `X_INVALID` - the client has provided invalid login credentials therefore re-authenticate.
- `X_AUTHENTICATED` - the login credentials provided by the client are correct and therefore the client has been authenticated.
- `X_ADIOS` - the client has successfully logged out of the system

The associated state diagram for a client connection via the `x_se_httpGateKeeper` is depicted in Figure 5-13. When a new request is received, the connection is either *rejected* if it's a blocked IP address or it is required to *authenticate* (i.e., redirected to an authentication portal). The users is then required to enter their credentials and if they are correct, they are *authenticated* or else an *invalidated.* The *authenticated* connections are then *let in* to make multiple requests on the system until they log out to end their usage session.

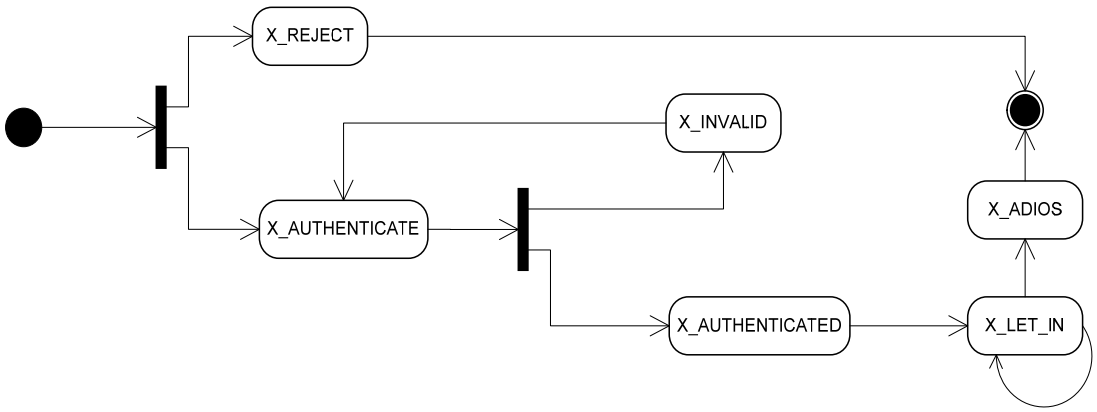

FIGURE 5-13 CLIENT CONNECTION STATE DIAGRAM

The authentication process for the client devices is discussed thoroughly in section 6.2.3.1 in the context of comparing it with the other authentication mechanisms currently being implemented for PALs.

5.5.2 Billing

The inherent extensible nature of the system poses a challenge as far as network usage accounting is concerned. This is due to the fact that there are various service element modules that can be implemented on the system and therefore there is no prior knowledge of the usage attributes that are revealed by each service element module. This creates a problem of semantic ambiguity associated with the usage metrics of the service elements because the billing modules are not aware of the meaning of the usage metrics and hence are unable to operate on the usage data. Semantic ambiguity generally arises when the meaning of a sentence or a word must be determined with the help of a greater knowledge sources (Baker et al, 1994). To illustrate the challenge posed by semantic ambiguity in SEHS, Table 5-2 shows the usage metrics associated

with a service element module. This information is accessible to the mediation system and consequently to the `x_bss` via the schema definition file that is associated with every service element module.

TABLE 5-2 HYPOTHETICAL USAGE METRICS

Category	**Usage attribute name**	**Data type**	**Presence**
Who	clientID	String	REQUIRED
When	startTime	Date/Time	REQUIRED
When	endTime	Date/Time	REQUIRED
What	quantity	Integer	REQUIRED

The meaning of this information however cannot be inferred by these system modules because the attributes could have different meaning depending on the service element module that is being implemented. For example, with `x_se_http` (the basic http service provisioning module) the attributes could have the following meanings Table 5-3.

TABLE 5-3 USAGE METRICS FOR HTTP SERVICE

clientId	The IP address of the requesting client
startTime	The time that the session started
endTime	The time that the session ended
Quantity	The number of requests made

And for the `x_se_http_gateKeeper` (the authenticated http service module), the metrics could have a different meaning Table 5-4.

TABLE 5-4 USAGE METRICS FOR AUTHENTICATED HTTP SERVICE

clientId	The username of the user on the network
startTime	The start time of the session

endTime	The end time of the session
Quantity	The amount of data download in that session

Ambiguity is an area that is well researched especially in the field of Knowledge Based Machine Translation systems where computers are used to translate documents from English into different target languages (Baker et al, 1994). It's a problem that arises where more than one interpretation is possible for a sentence. The different types of ambiguity that arise in language are lexical ambiguity, syntactic ambiguity and semantic ambiguity. There are different approaches that are normally used for disambiguation[5] in machine translation systems:

- Constraining the lexicon - this is done by limiting the words and phrases in the language to those words that do not exhibit lexical ambiguity.
- Constraining the source language construction - this involves the elimination of syntactic and semantic ambiguity by defining a set of rules, or a controlled grammar that guides the language sentence construction.
- Interactive disambiguation - In the cases where automatic disambiguation is impossible, human interaction is consulted to assist in the disambiguation of the sentences.

These above mentioned approaches are specifically for disambiguation in the field of machine translation systems. Close parallels can be drawn though with regards to solving the challenge of ambiguity in SEHS. Applying the first approach, constraining the lexicon, in SEHS would imply having to define a set of attributes or tokens that would commonly be used by all SEHS service element modules to define the associated network usage metrics. For example, a set of usage attributes say (e.g., `clientID`, `clientIP`, `startTime`, `endTime`, `numOfRequests`, `sessionID`) would have to be defined and each of these metrics would have to be described. This would essentially eliminate having different service element modules using different metrics for same usage attribute (i.e., the `clientID` metric would be used exclusively to represent a username for the client using the network service and not describe a client device's IP). This approach would in a sense allow the billing

[5] The process of reducing or eliminating ambiguity

modules to have a pre-knowledge of the usage metrics and hence the ability to operate on the usage data. There are a few drawbacks associated with this approach:

- It would require a very large base set of pre-defined usage attribute descriptors.
- It would necessitate an update and version control mechanism of the attribute descriptors as new service element modules, with new and different usage attributes, are developed.
- It would necessitate standardization of the attribute descriptors so that all the service element modules and billing modules developed are using the same standard descriptors.

The approach that is implemented in SEHS for disambiguation of the service element usage metrics is interactive disambiguation. This is done by allowing for the dynamic configuration of the billing modules based on the underlying service element usage attributes. It essentially allows for an attribute mapping to be done between billing module variables and the service element module attributes. Figure 5-14 shows an attribute mapping between a simple usage based billing module and a VoIP service element module. The variables that are need by billing module to complete the billing transaction are a `clientID`, minimum charge and unit cost (which are supplied by the user) and `unitQuantity`. The VoIP module on the other hand records a large number of usage attribute associated with a VoIP service but for the purpose of usage billing, only `subscriberID` (which is mapped to the `clientID` variable in the billing module) and the `callDuration` (which is mapped to `unitQuantity` variable) are essential.

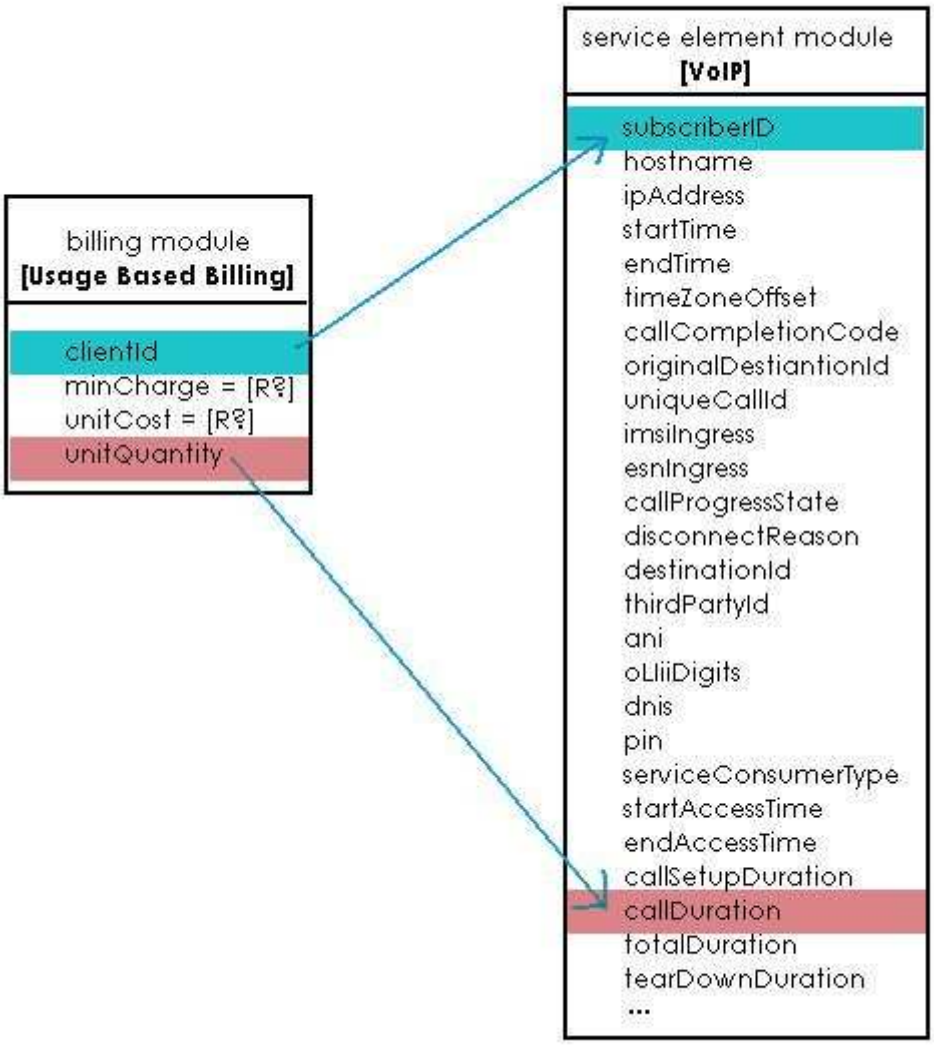

FIGURE 5-14 ATTRIBUTE MAPPING FOR A VOIP SERVICE

Figure 5-15 shows the attribute mapping between the usage based billing module and a different service element module (i.e., a simple http module). In this particular case, the `clientID` and `unitQuantity` variables in the billing module map on to the `clientID` and the `dataDownloaded` usage metrics in the service element module respectively.

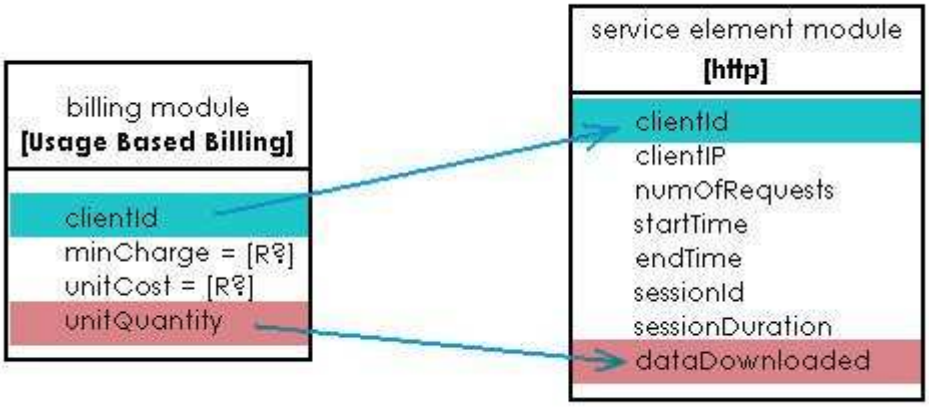

FIGURE 5-15 ATTRIBUTE MAPPING FOR A HTTP SERVICE

When a billing module is configured, it queries the underlying service element module for the schema definition file (Listing 5-4). It then extracts the usage metrics from the schema definition file and allows the user to map the billing module variables with the associated billing module attributes (Figure 5-16).

```
private void schema_attributes(String schema_name)
 {ArrayList attributes = new ArrayList();
  try
   {Descriptor descriptor = new Descriptor();
    String[] arb = new String[1];

    attributes = descriptor.parseSchema(schema_name, arb);

    attributeNames = new String[attributes.size() / 3];
    attributeTypes = new String[attributes.size() / 3];

    for(int a = 0; a < (attributes.size() / 3); a++)
     {attributeNames[a] = (String)attributes.get(a * 3);
      attributeTypes[a] = (String)attributes.get((a * 3) + 1);
     }
   }
  catch(org.ipdr.utils.IPDRException e)
   {System.err.println("IPDR exception : " + e.getMessage());
   }
  catch(org.xml.sax.SAXException e)
   {System.err.println("SAX XML exception : " + e.getMessage());
   }
  catch(java.io.IOException e)
   {System.err.println("IO exception : " + e.getMessage());
   }
 }
```

LISTING 5-4 QUERYING THE SCHEMA DEFINITION FILE

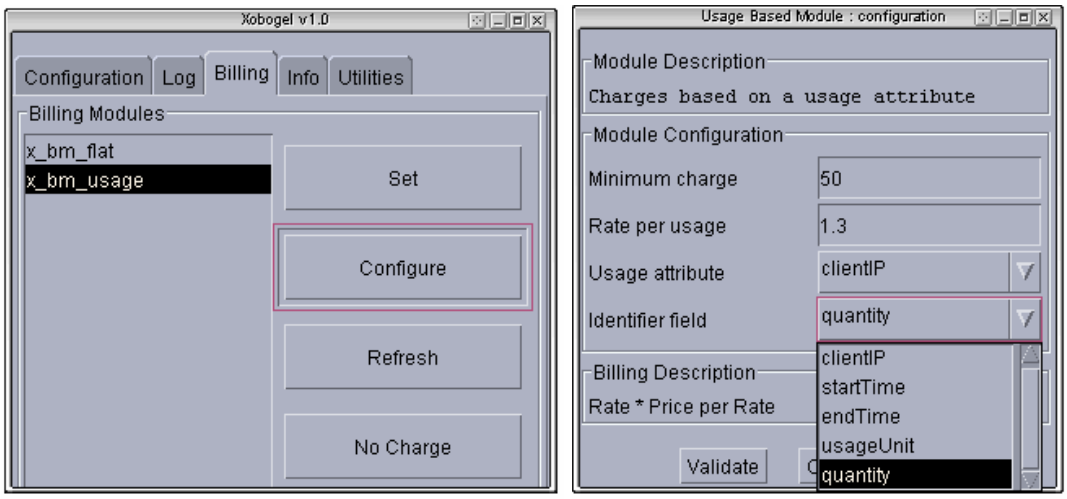

FIGURE 5-16 CONFIGURING USAGE BASED BILLING MODULE WITH X_SE_HTTP

Figure 5-17 depicts the same usage based module (as in Figure 5-16) being configured for the `x_se_httpGateKeeper` service element module. It highlights the different usage metrics that have been defined for the `x_se_httpGateKeeper` module versus those defined for `x_se_http` in Figure 5-16**.**

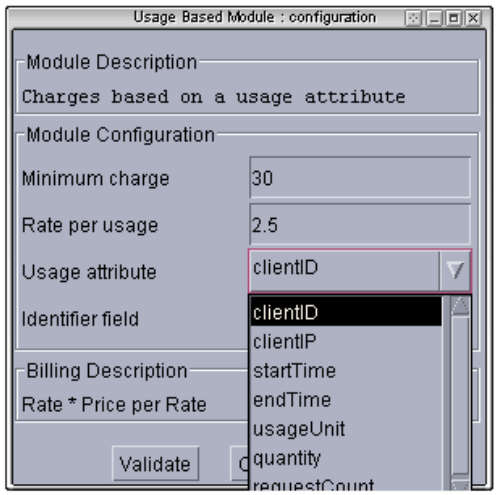

FIGURE 5-17 USAGE BASED BILLING MODULE FOR X_SE_HTTPGATEKEEPER

This functionality, of interactive disambiguation through attribute mapping, allows the different billing modules to be used in conjunction with different service element modules. In order for a module to be operational on SEHS, it has to implement a

simple `x_billModule` interface (Listing 5-5), that defines the methods that are called by the `x_engine` to perform the billing transaction.

```
import java.util.*;

public interface x_billModule
{public void configure(String se_xsdname);
 public String description();
 public void charge(ArrayList usage);
 public void reset();
 public String formula();
}
```

LISTING 5-5 BILLING MODULE INTERFACE

A brief description of the methods defined in the interface follows:

- `Configure(String se_xsdname)` is the method that is called to configure the billing module. It is passed the schema definition file associated with the service element module as a parameter.
- `Description()` is a textual description of the operation of the billing module.
- `Charge(ArrayList usage)` is the method that is called to compute the charge associated with usage data. The method is passed the usage data as a parameter, it then extracts the required variables, based on the configuration settings and calculates the charge. This method then passes the charge information to the `x_bss` to handle the business activities related to billing users.
- `Reset()` is used to reset the settings of the billing module.
- `Formula()` is used to provide a textual description of the formula that is used in the calculation of the charge.

5.5.3 Coupon Billing

Section 3.1.3.3 highlighted that the billing method most preferred by small businesses is the pre-paid as opposed to post-paid billing. In order to facilitate the implementation of pre-paid mechanism on the PALs, a coupon billing infrastructure

has been implemented. This allows users to go to a PAL, buy a coupon for a specific usage quantity (e.g., for a specific time, or data downloaded), and use the coupon ID to authenticate.

Coupon Billing is implemented in a service element module called `x_se_httpCouponKeeper`. The coupon billing infrastructure is implemented using a different mechanism from the post-paid billing because the need to maintain the persistence of usage accounting information for pre-paid billing is more minimal than for post paid. The coupon billing infrastructure implemented bypasses the functionality of the mediation system because the IR, IS and IT functionality that it provides is not an absolute requirement and also for the sake of maintaining simplicity in the system.

One of the utilities provided as part of the coupon billing infrastructure is the coupon generation functionality (Figure 5-18). This allows the WISP owner to generate two kinds of coupons (i.e., time-based coupons (Figure 5-19) and data-based coupons (Figure 5-20)). These correspond to the usage based billing scheme options of charging per time spend using the service or the amount of data downloaded during the service usage respectively.

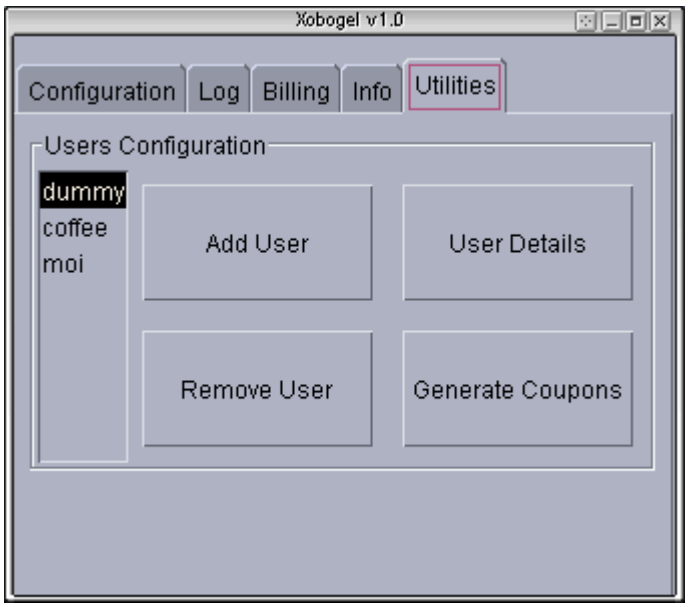

FIGURE 5-18 GENERATE COUPONS FUNCTIONALITY

The actual coupon IDs are generated using a simple algorithm (Listing 5-6) that calculates the coupon Id using temporal metrics associated with the time of creation of the coupon and also introduces a random seed into the calculation. While this algorithm is not adequate for a full functioning production scale system, it is sufficient for testing purposes on the system and for the proof of the concept of embedding the coupon billing infrastructure into a service element module.

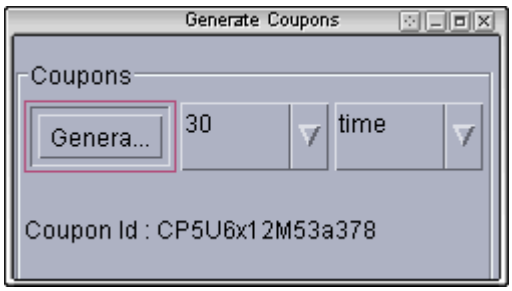

FIGURE 5-19 GENERATE A TIME BASED COUPON UTILITY

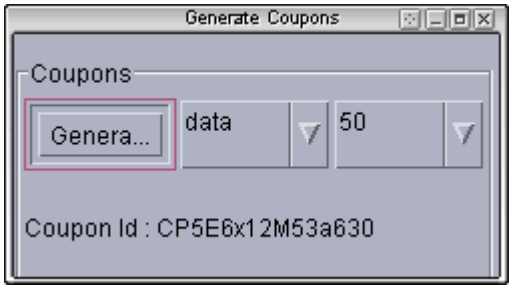

FIGURE 5-20 GENERATE A DATA BASED COUPON UTILITY

```
Calendar cal = Calendar.getInstance();
String arbword = "SEHSCOUPONSEED";
int rnd = (int)(Math.random() * arbword.length());

int month = cal.get(Calendar.MONTH);
int day = cal.get(Calendar.DAY_OF_WEEK);
int hour = cal.get(Calendar.HOUR_OF_DAY);
int min = cal.get(Calendar.MINUTE);
int sec = cal.get(Calendar.SECOND);

int thetime = 0;
int thedata = 0;

String id = "CP" + month + arbword.charAt(rnd) + day + "x" + hour +
                                "M" + min + "a" + (sec * arbword.length());
```

LISTING 5-6 COUPON ID GENERATION

Once the coupon has been generated, it is immediately stored in the system database and it is then used for authenticating the users on the system. The coupon ID is used

as both the username and the password (Figure 5-21). During the user's session on the network, the system keeps a check on the amount of time spent on the network or the data downloaded depending on whether the coupon is time-based or data-based respectively.

FIGURE 5-21 AUTHENTICATION WITH A COUPON ID

The `x_se_httpCouponKeeper` introduces three other connection states, to the five already defined states, that are used in the provisioning of coupon billing (Listing 5-7):

- `X_EXPIRED` - indicates a connection that has expired due to the coupon value having been exhausted
- `X_CP_AUTHENTICATED` - indicates a client connection that has been authenticated via the coupon infrastructure
- `X_CP_ADIOS` - this is a coupon authenticated connection that has logged out of the system. When the clients log out of the system, the coupons are updated to reflect the remaining value.

```
/** the session has expired */
public final int X_EXPIRED = 7;
/** authenticated the coupon login */
public final int X_CP_AUTHENTICATED = 8;
/** logged off the coupon */
public final int X_CP_ADIOS = 9;
```

LISTING 5-7 X_SE_HTTPCOUPONKEEPER CONNECTION STATES

5.5.4 Automatic Proxying

There is underlying need in the usage of network services for simplicity and ease of use. In the context of WiFi hotspots this translates to the ease of associating with an access point (i.e., configuring the device to use the access point, the ease of logging onto the network, and the ease of using the network services). Configuring a mobile device to use a wireless access point can be a non-trivial task to a vast majority of PAL users who are not technically competent. Seamless[6] usage of the network services would benefit these users and improve the usability of the PAL. The typical device reconfiguration that needs to be done to get on to a hotspot may include:

- Setting an IP address that is within the subnet of the access point. This goes along with setting the subnet mask, and the default gateway to all IP communication from the device.
- Setting up proxies servers for access to resources (e.g., web proxy server address and port for web browsing).

One of the solutions currently available for alleviating some of the device reconfiguration task is using Dynamic Host Configuration Protocol (DHCP) (Hunt et al, 1998). DHCP solves the problem of configuring clients IP addresses and other IP configuration information. It doesn't however handle the configuration of web proxy servers because this is done at the application (web browser) level. This therefore necessitates a solution that would allow for handling of client requests without having to do any reconfiguration on the client device. A typical HTTP transaction (discussed in Appendix C: A Web Browsing Transaction) involves performing a DNS resolution to determine the IP address associated with a DNS name and then performing an ARP resolution to determine the MAC address associated with an IP address. Intercepting client requests can be done at each of these two steps.

[6] denotes the ability to get on the network without having to reconfigure their devices

5.5.4.1 Intercepting the Client's Request

The commencement of the HTTP request only happens after the three-way handshake has occurred between the web server and the requesting machine. That depends on the MAC address of the web server having been determined from ARP which resolves the IP address received from DNS.

Intercepting the client's request basically involves spoofing the identity of the web server such that the requesting host sends the requests to the intercepting host, which in turn processes the request on behalf of the originating host.

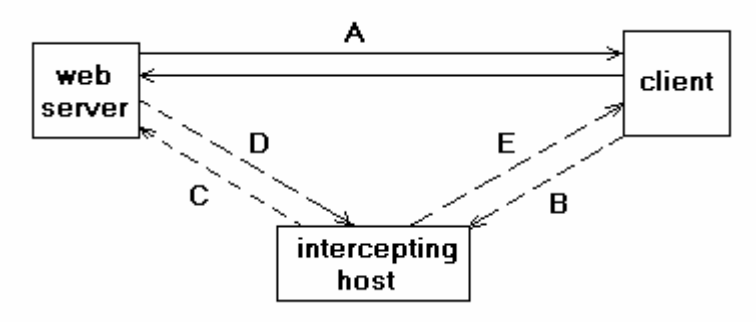

FIGURE 5-22 INTERCEPTION HTTP COMMUNICATION

'A' REPRESENTS THE NORMAL 'UN-INTERCEPTED' COMMUNICATION BETWEEN THE CLIENT AND THE WEB-SERVER. WHEN THE CLIENT'S COMMUNICATION IS BEING INTERCEPTED, THE REQUESTS ARE SENT TO THE INTERCEPTING HOST (B), WHICH IN TURN FORWARDS THE REQUEST TO A WEB-SERVER (C). THE WEB-SERVER THEN RESPONDS TO THE INTERCEPTING HOST (D) WHICH IN TURN FORWARDS THE RESPONSE TO THE CLIENT (E). IN ESSENCE THE INTERCEPTING HOST ACTS AS A TRANSPARENT WEB PROXY TO THE CLIENT

In order to get the client to send requests to the intercepting host, the client has to 'think' that the intercepting host is the web-server that it needs to query. The two main ways that the client determines the identity of the web server is first by performing a DNS request, to determine the IP address and by performing an ARP request to determine the MAC address of the web-server. Consequently the two ways that can be implemented to intercept the client request are via spoofing an IP address (during the DNS resolve process) or spoofing a MAC address (during the ARP process).

5.5.4.1.1 Spoofing the IP address

DNS poisoning is when a fake IP address is send in response to a DNS request from a client. For example, if a client sends a DNS request for www.cs.ru.ac.za, the DNS server would respond with 146.231.120.71, the IP address of spiderman.ict.ru.ac.za (www.cs.ru.ac.za is an alias for spiderman.ict.ru.ac.za). In order to intercept the requests, a DNS response could be constructed with the IP address of the intercepting host (e.g. 146.231.123.12) and then send back to the client. For all communication purpose the client supposes that 146.231.123.12 is the IP address for www.cs.ru.ac.za and so it continues to resolve the MAC address from the IP address, executes the three-way handshake and then continues with communication. The intercepting host, 146.231.123.12, then connects to the web-server and communicates the web response to the client.

5.5.4.1.2 Spoofing the MAC address - (ARP poisoning)

ARP poisoning occurs when a reply to an ARP request is a fake MAC address. As opposed to a DNS request, where there is a dedicated DNS server that handles DNS requests, ARP relies on the individual hosts responding to the ARP request broadcast. When a host receives an ARP request, it checks its IP address to determine if it's the requested one. If that is the case, the host then responds to the ARP request by sending a reply with its MAC address. Since the ARP request is a broadcast i.e. send to ff:ff:ff:ff:ff:ff MAC address, it is received by every host on the network. The intercepting host also receives the ARP request and so it also constructs an ARP response containing its MAC address and sends it to the client. On receipt of the ARP response, the client updates its ARP table, executes the three-way handshake and continues with communication.

5.5.4.1.3 Other request interception methods

Other methods of providing client request interception functionality are:

- Hardware based solutions: these are normally vendor specific implementations that provide a gateway device that intercepts all unauthenticated client

requests and forwards them to an authentication page. Examples include the Nomadix Gateway[7], Cisco BBSM-Hotspot[8] and the Aptilo Access Gateway[9].

- NAT – (Network Address Translation) is where client requests are mapped onto a different network address, essentially redirecting the clients' requests to a different address. An example of this functionality is provided in linux by iptables and it allows one to define a rule in the NAT table using the PREROUTING chain. The rule simply does a DNAT based on the protocol (TCP) and the port that's used for incoming client request on the SEHS PC.

The availability of these different options that can be used for increasing the usability of the PALs leaves the final decision with the individual WISPs. The choice of the method to implement is influenced to a large extend by the costs associated with implementing that solution and also by the level of computer expertise of the PAL users.

5.5.5 SEHS Backend

The backend component of the system is responsible for maintaining the persistence of the data objects that are used by SEHS. The main data objects that are used by the system are:

- The USERS table which is responsible for keeping track of the PAL user details. The information that is stored in this table is the username and password of the users, the account balance which reflects the amount that is owed by the user for the network usage and the IP address field that is used for the purpose of validating the users' identity.
- The COUPONS table that is responsible for keeping track of all the coupons that have been generated for use on the system. This table keeps track of the ID, the type (data or time) and the value of the coupons. This table is used particularly by the service element that provides the functionality to authenticate of the system using coupons.

[7] http://www.nomadix.com/products/platforms/hsg/

[8]http://www.cisco.com/en/US/products/sw/netmgtsw/ps533/products_user_guide_book09186a0080126bfe.html

[9] http://www.aptilo.com/solutions_public_access.htm

- The other need for persistence in the system is with regards to the IPDRDoc (i.e., usage records, which need to be stored). This functionality is implemented in the mediation system component of SEHS. The mediation system encodes the usage data from the service elements and then stored this information to a local storage medium and it makes this information available to request components via a communication protocol that is established for the exchange of usage information between the different system/business units.

The coupons and the users' information are stored on a MySQL DBMS as simple database tables. The main advantage provided by MySQL in context of small PAL management systems is with regards to the costs associated with acquiring the software. MySQL is freely available[10] under the Open Source License (MySQL, 2004) which allows for the use of the application under the condition that the application that is utilizing it is open source as well[11].

5.6 Summary

This chapter provided an overview of the SEHS system by detailing the data objects and the sequence diagrams associated with the usage of the system. The extensibility of the system is also highlighted with a detailed example of implementing a service element module (`x_se_http`). The example also discusses how usage accounting is achieved in SEHS and how it is facilitated by the IPDR standard. The functionality that is implemented in the system is discussed with a specific focus on the billing infrastructure, interactive disambiguation of the usage metrics, and the coupon billing functionality.

[10] Downloads available at http://sunsite.mff.cuni.cz/MIRRORS/ftp.mysql.com/downloads/index.html
[11] The full licensing policy can be viewed at http://www.mysql.com/company/legal/licensing/

6 Chapter 6: SEHS Experimentation and Discussion

The system that has been implemented sought to meet the specific requirements that were determined in chapter 3. Various experiments and test have been undertaken on the system to determine the applicability of the system and the framework in meeting the identified requirements and design goals. The experiments performed provide a conclusive validation of the system and an indication of the successfulness of the research. The experiments are discussed in detail herein after and the specific features of the system are also detailed. This is done by juxtaposing the functionality achieved in SEHS with the functionality available in other PALs management systems that are currently available.

6.1 Load testing SEHS

The experiments run on SEHS were to test the specific areas and functionality of the system in terms of how they meet the requirements that have been previously highlighted. This is in terms of both the users' and the WISPs' expectation of the system (e.g., performance expectations, availability expectations).

The experiments were executed with the SEHS running on a PC with the specifications mentioned in section 5.1.2. The HP Omnibook XE_3 was the wireless client that was used to generate the requests on SEHS. A Buffalo 802.11g wireless PCMCIA card was used for connecting the laptop to the AP.

6.1.1 Aim

The first experiment performed on SEHS was for the purpose of stress testing the application to determine how it handles increasing user loads. In Chapter 3, the performance and availability expectations of the users were highlighted. Through this experiment, it can be ascertained how the system performs and depending on the specific metrics assimilated from user requirements, it can be concluded if the system is applicable in meeting those user requirements. The results of this experiment also helped in predicting and pre-empting the behaviour of the system under different loads and also in determining the reliability of the system as far as increasing usage density is concerned.

6.1.2 Methodology

The tool that was used in executing this experiment is Proxy Sniffer™[12] web load testing tool. Proxy Sniffer™ operates by first recording a web surfing session, after the session has been completed, it is saved and then a test script is generated. The recorded session script is then used to simulate a scenario where a specified number of users are executing the recorded session.

The experiment was designed to simulate the action of users requesting new connections to the system. The number of simulated users ranged from 1 to 20. This range is representative of and above the expected number of small PALs users at any one time period (see Section 3.1). The upper limit of 20 users in the experiment allows for an observation of how the system operates under normal usage (i.e., ten users) and how it scales and performs to an increase in the number of users (i.e., 20 users). The experiment simulated requests on SEHS for a resource that was stored locally on the SEHS PC. In undertaking the experiment, the aim was to emulate as much as possible the typical usage of the hotspot system (i.e., in terms of the number of users and the nature of the usage session). The number of user has already been defined to be approximately 10 and it has been ascertained as well that one of the top uses of the Internet is web browsing. Studies that have been undertaken suggest that the typical average size of web pages is 60 KBs (Baer N., 2000) and for the purpose of this experiment, a number of web pages were evaluated in order to validate the average size claim. The websites that were profiled are online news websites (assuming that they are representative of a typical user's browsing session) that were chosen randomly (Table 6-1). The average total webpage size was determined to be 182 KB.

TABLE 6-1 RANDOM NEWS WEBSITES PROFILE

name	URL	Page	Webpage Files	Total
abc news	http://abcnews.go.com/	51	139	190
bbc news	http://www.bbc.co.uk/	34	36	70
cbs	http://www.cbs.com/	87	123	210

[12] Available at http://www.proxy-sniffer.com

cnn	http://www.cnn.com/	52	153	205
fox	http://www.foxnews.com/	83	175	258
gnews	http://news.google.com/	130	76	206
google	www.google.com	21	8	29
hotmail	www.hotmail.com	19	31	50
iol	www.iol.co.za	96	55	151
mng	http://www.mg.co.za/	123	86	209
msnnews	http://www.msnbc.msn.com/	68	133	201
news24	www.news24.co.za	105	104	209
sapost	http://www.southafricapost.com/	83	257	340
usatoday	http://www.usatoday.com/	81	184	265
wired	http://www.wired.com/	33	50	83
yahoo	www.yahoo.com	55	38	93
ynews	http://www.foxnews.com/	237	94	331

Average		**79.9**	**102**	**182**
smallest		**19**	**8**	**29**
Largest		**237**	**257**	**340**

In this experiment, the file that was requested from the server was made to be larger than the expected average webpage i.e. the file is 1000 KB large instead of 182KB. This is simply to push the system to higher usage levels in order to observe how it deals with the intensity of handling larger files and more connection requests. The assumption behind this is that if the system can handle the larger files and more users efficiently, then it would be able to handle normal usage and to scale to higher usage levels.

The parameters of the experiment are detailed below (Table 6-2).

TABLE 6-2 STRESS TEST PARAMETERS

Number of concurrent users	1 – 20
Loops per user	10
Applied HTTP protocol version	1.0
Start-up delay per user	0 milliseconds
Request Timeout per URL	60 seconds

6.1.3 Experiment results and discussion

Once the experiment parameters had been set, Proxy Sniffer™ executed the test resulting in the following observations (Table 6-3).

TABLE 6-3 STRESS TEST RESULTS

Number of Users	**1**	**5**	**10**	**15**	**20**
Total URL calls: passed	10	50	100	150	200
Total URL calls: failed	0	0	0	0	0
Average network connect time (ms)	8	12	19	17	25
Recycled Network connections	0.0%	0.0%	0.0%	0.0%	0.0%
Average Session Time per user/loop (sec)	**0.53**	**2.54**	**5.08**	**7.84**	**10.80**
URL response time per user (sec)	0.53	2.54	5.08	7.84	10.80
Average response time per page (sec)	0.53	2.54	5.08	7.84	10.80
Network throughput per user (kbps)	2111.71	439.83	219.35	142.25	103.29
Web transaction rate (calls/sec)	1.65	1.84	1.84	1.82	1.75
Session failure rate	0.0%	0.0%	0.0%	0.0%	0.0%
Total network throughput (kbps)	1843.86	2056.48	2057.05	2029.16	1949.17

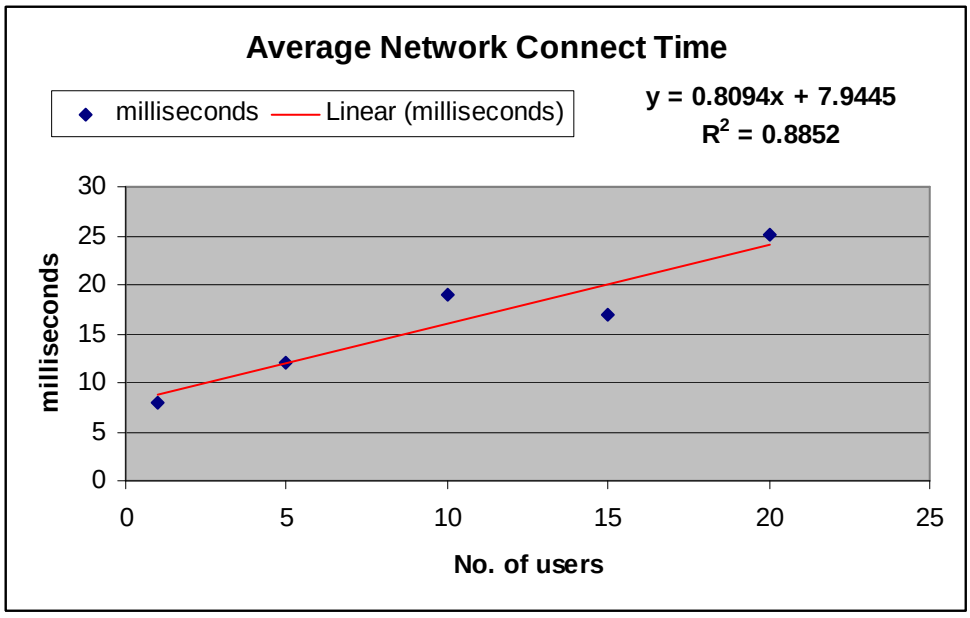

FIGURE 6-1 AVERAGE NETWORK CONNECT TIME

The trend depicted in Figure 6-1 is the time it takes to establish a connection to the server. This is the time that is spent before any communication between the server and the client is initiated. As the number of users increases on the PAL, the time it takes for the connections to be established also increases (e.g., 12 ms for 5 users and 17 ms for 15 users). The network connect time adds to the perceived performance of the server, from the users' point of view, in that the longer it takes for the connections to be established the slower the server is perceived to be and vice versa. The time to connect to the server has a linear relationship ($y = 0.8094x + 7.9445$, $r^2 = 0.8852$ and p-value = 0.017) (Figure 6-1) to the number of users because an increase in the number of users means an increase in the number of processes which are handled by limited system resources. A unit increase in the number of users corresponds to approximately a 0.8094 millisecond increase in the average network connect time. This corresponds to a steady, gradual performance degradation associated with increasing number of users as far as network connect time is concerned.

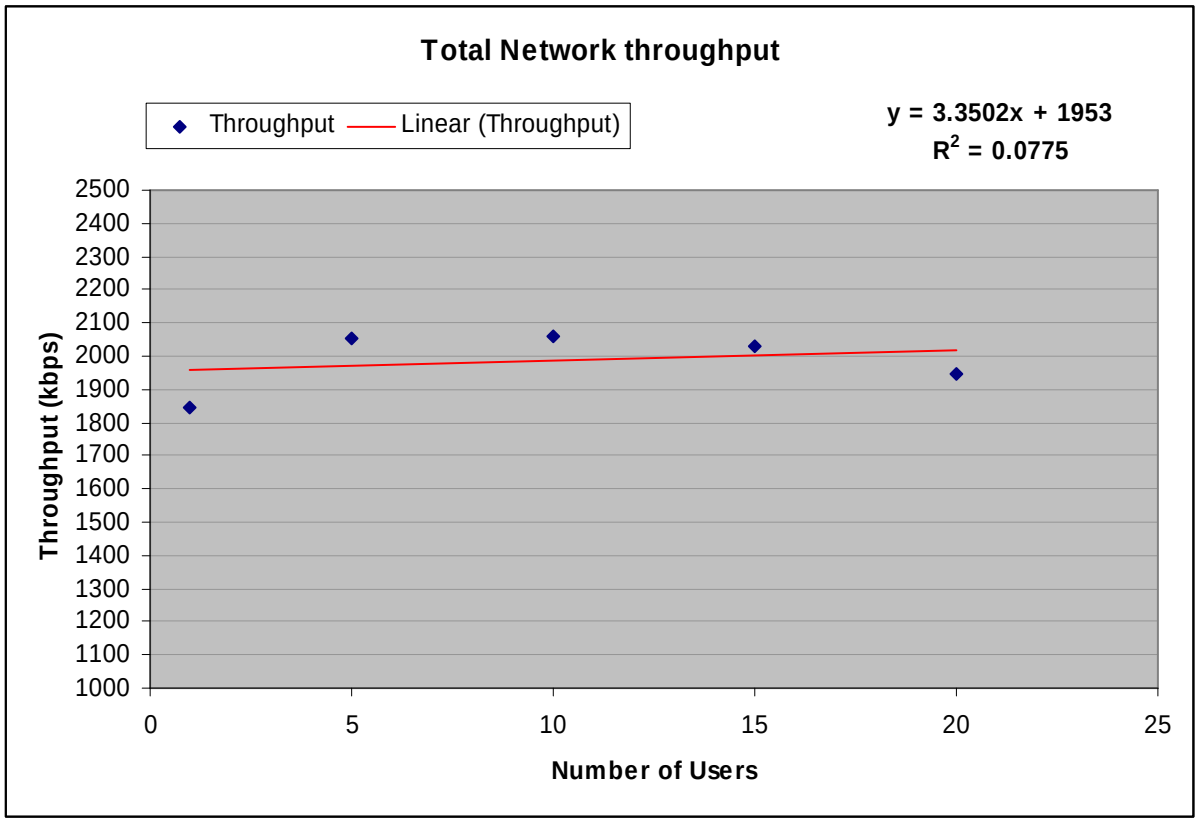

FIGURE 6-2 TOTAL NETWORK THROUGHPUT

The metric of concern as far as meeting the performance expectations of the users, is the throughput available on the network. Figure 6-2 shows a scatter graph of the total network throughput associated with different usage loads. This gives a more accurate indication of the overall performance of the system than the throughput per user per session metric because it takes into consideration the total network throughput associated with all the users on the system. The throughput level attained on the system works out to approximately 198.7 Kbps[13] which is above the 100 Kbps per user range that is recommended by Intel (Chapter 3).

One of the metrics that influence the users' experience of the system is the average session time per user per loop. This metric shows the average time that it takes for each emulated user to complete a usage loop on the system. A loop in this case is a pair of HTTP request and response messages between the client and the server. This metric is the observable system performance from the perspective of the users. Figure 6-3 indicates the average session time and the best fit trend line drawn through the points. The trend line shows a linear relationship ($y=0.5389x - 0.1385$, $r^2=0.9988$, $p=0.000018$) between the number of users and the average session time. The average session time increases (by 0.5389 seconds per user) as the number of users increases.

[13] An average of the observed total network throughput divided by the number of users i.e. (1843 + 2056.48 + 2057.05 + 2029.16 + 1949.17) divided by 10.

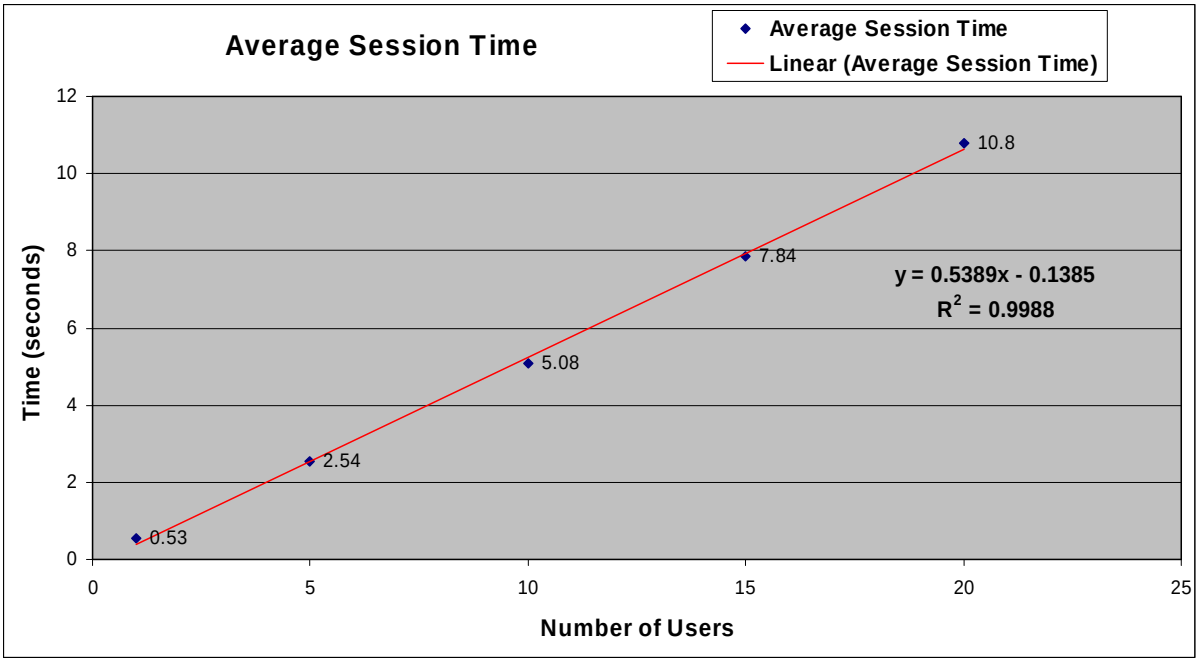

FIGURE 6-3 SCATTER GRAPH AVERAGE SESSION TIME

Within the range of 1 to 20 users, the system's performance as far as average session time per user is concerned is linear. From the R^2 value associated with this model (Figure 6-3), it is observed that 99.8% of the variation in the system's performance (in terms of average session time) is attributable to the linear relationship with the number of users. The linear relationship (versus exponential or logarithmic) indicates that the system is able to scale well to an increase in the number of users.

Another experiment was run on the system to support the findings about the average session time per user. The aim of this experiment is to observe the change in the average session time per user over time. The experiment was setup with 10 emulated users each with 50 (the maximum number of session loops available on the test application) session loops. The number of users is fixed at 10 because that is the assumed usage density on PALs (section 3.1).

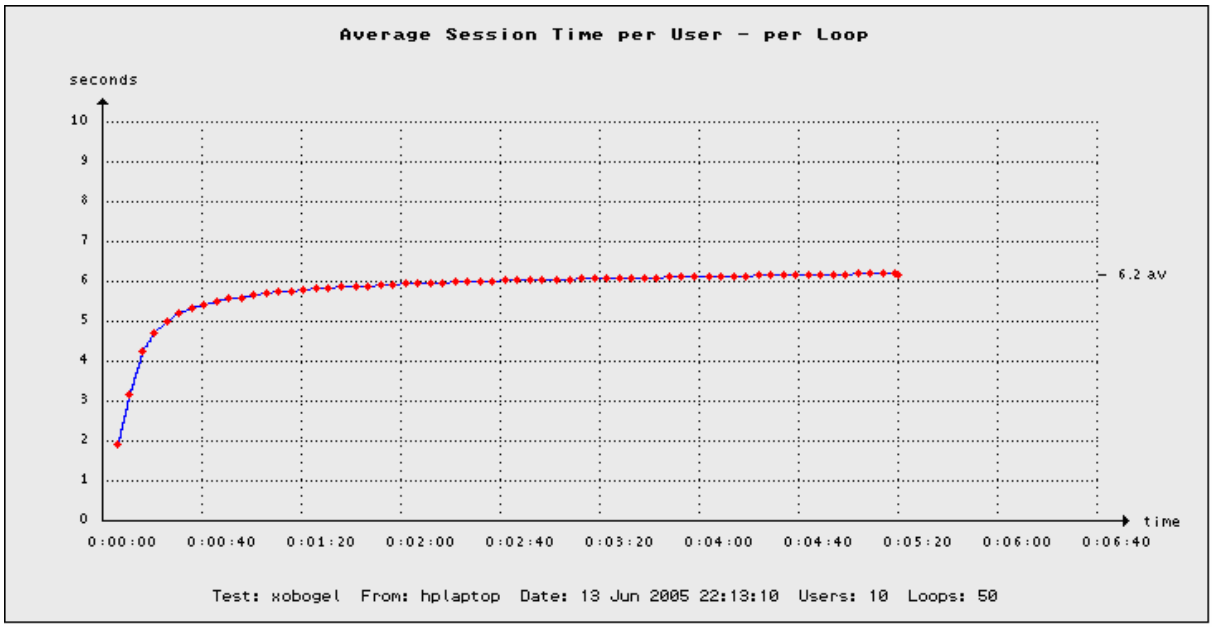

FIGURE 6-4 AVERAGE SESSION PER TIME PER USER FOR 10 USERS, 50 LOOPS

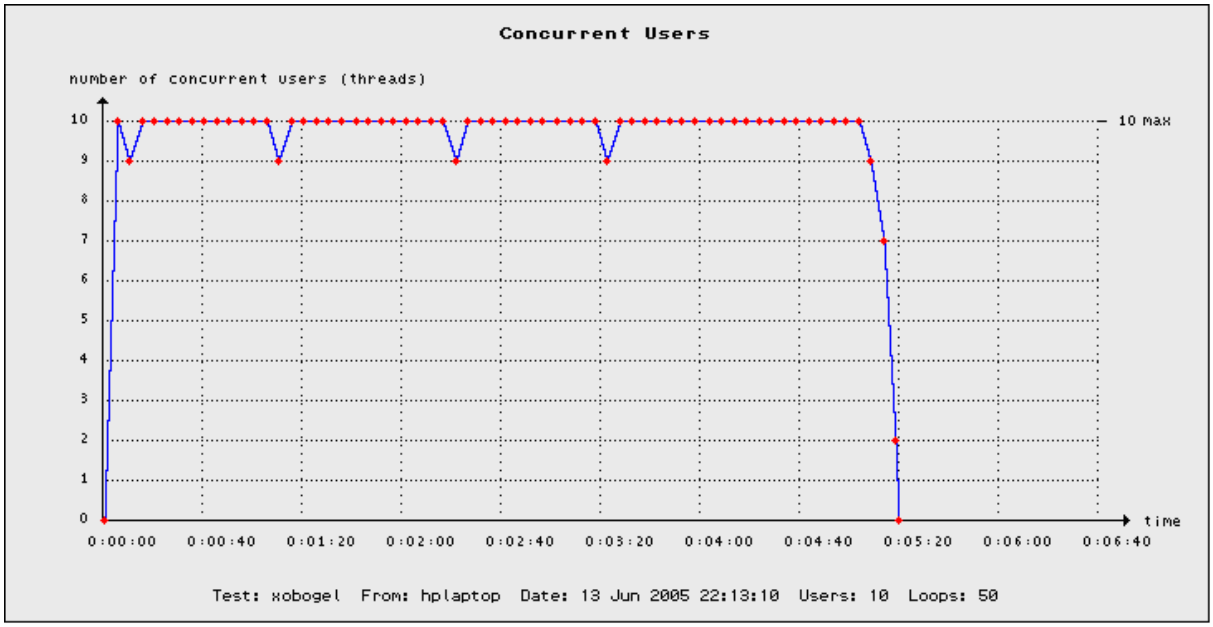

FIGURE 6-5 NUMBER OF CONCURRENT USERS PER TIME

The observed average session time per user per loop averages out and balances at approximately 6.2 seconds (Figure 6-4). This can be assumed to be the average session time per user over the period of the system uptime. The throughput per user per loop can be worked out as follows:

Data downloaded per session loop = *1000 KB*
Average session time per user per loop = *6.2 sec*
Therefore, *1000KB / 6.2 sec = 161.3 KBps*

Average throughput = *161.3 KBps*

The average throughput per user, at this higher usage levels (i.e., 10 users (Figure 6-5) at 50 loops per user) still works out above the recommended throughput (i.e., 100Kbps (Chapter 3)) as far as the user's expectations are concerned.

The above results indicate the efficacy of the system in meeting the performance expectations of the users. This is despite the fact that the experiment is performed under increased usage loads (i.e., a larger requested resource and number of users) than the normal operating conditions of small PALs. As far as system availability considerations are concerned, the system has been tested at a usage load of 20 concurrent users which is twice the anticipated usage load on small PALs (section 3.1), and the average session time per user per loop has been positively linear with 0% failed connection errors. This indicates the applicability of the system and its appropriateness for small PALs as far as meeting the performance and the availability expectations associated with small PALs environment.

6.2 SEHS vs. Other PAL management systems

SEHS has been designed to meet the specific requirements of small PALs and to provide a solution that allows for an extensible, easily deployable and simple PAL management system. A number of systems and solutions exist that also seek to provide the needed functionality of managing PALs. The functionality provided by these systems is detailed below and subsequently compared with the functionality implemented in SEHS in terms of the specific requirements that have been determined for small PALs;

6.2.1 LESS Networks Hotspot Server

LESSnetworks[14] have developed a hotspot server that manages public provisioning of free wireless service. The LESSnetwork server is distributed as a bootable CD or a

[14] http://www.lessnetworks.com/

single .iso[15] file from which a bootable CD can be created. Installing the Hotspot server on a machine completely reformats the hard drive of the computer and installs the server components. The Hotspot server (version 0.99) is based on Redhat 8.0 operating system and NoCatAuth (section 6.2.3) software. System requirements for running the Hotspot server are:

- Processor - 133MHz Pentium
- RAM - 64 Mb
- Hard Drive - 1Gb
- Networking - 2 NIC cards
- CD-ROM - required

The functionality provided by the Hotspot server is detailed below (LESSnetworks.com, 2005):

- To Users
    - One world wide login credentials
    - Consistent, reliable free WiFi
    - Venue-to-venue chat
    - Promotions
- To Venues
    - Service branding with logo and website
    - Functionality to email clients
    - Access to usage statistics
    - Sell advertising space
- To WISPs
    - Manage multiple locations via a web interface
    - Diagnostics tools to boost traffic and to respond to trouble
    - A trouble-ticked system to track issues in large multi-venue deployments
    - Monitoring and alerting functionality
- Other
    - Runs NoCat in passive mode for authenticating to the server

[15] ISO-9660 filesystem image file extension

- Runs a cronjob to periodically ping the central server with information to keep the hotspots listing updated
- Downloads and installs software updates from the central server
- Runs a DHCP for the local clients
- Centralized authentication, management and security functions

One of the main advantages of the LESSnetwork hotspot server is the fact that is it available on the GNU Public License (GPL) (GNU, 2005) license hence it is freely available for businesses that want to implement a hotspot service. In terms of meeting the requirements for small PALs, the distribution licence provides the initial justification of the applicability of the Hotspot server in terms of reduced cost of acquiring and deploying the system. The cost expectation considerations are both for the WISPs in terms of how much they are prepared to invest into the hotspot and also for the users in terms of the cost of the usage of the service. A majority of the users indicated a strong desire for free internet billing scheme (section 3.2) and hence by providing an application that is freely available and that handles the provisioning of free internet access, the LESSnetorks server is set to capture the initial stakeholders cost preferences more accurately. The server is targeted specifically towards free hotspot provisioning. This can be a disadvantage because it puts a limitation on the WISPs in terms of the business model that they can implement on the hotspot.

The Hotspot server is based on the Linux operating system. The study undertaken in section 3.1.3.1 indicated a strong propensity among the businesses to use MS Windows operating system. This is normally the operating system that is implemented for the business support systems. Installing the Hotspot server formats the hard drive of the PC used and installs the RedHat operating system. This becomes a drawback as far as small WISPs are concerned because it entails having to set aside or acquire one PC for hosting the server and also acquiring the expertise that might be required to administer the system. These factors might increase the total cost of ownership of the hotspot. LESSnetwork has a network of partners who are able to provide technical assistance although these are concentrated around the deployments in the United States of America.

Third party administration of the Hotspot server is advantageous as far as providing a central portal for management of the available hotspots and for allowing for centralized security updates to be done. It however reduces the autonomy of the business implementing the hotspot and it might not necessarily be the desired arrangement for every business. Using third party authentication servers would be an advantageous arrangement in the light of roaming relationships between different hotspots because it would make authenticating users an easy operation. In cases where the business is a stand alone PAL that offers free internet access however, third party authentication servers are not as beneficial.

6.2.2 PublicIP ZoneCD

ZoneCD is a bootable live CD that allows for creation of an automatic WiFi gateway. The ZoneCD is a product that brings together different programs that have been written and implemented to provide different functionality needed for network services provisioning. It is based on the Knoppix[16] operating system and implements the NoCat authentication infrastructure. HTTP service is provided and facilitated by the Apache[17] web server and the Squid[18] proxy server for caching. DansGuardian[19] is used to provide the web content filtering functionality which operates in conjunction with the squid proxy server. The ZoneCD can operate in two modes, the open mode and the closed mode. In the open mode of operation, the following features are provided (PublicIP, 2005):

- Homepage redirection
- Customize ZoneCD splash page
- Content Filtering
- Customized Firewall rules

In the closed mode, the features that are provided are:

- User authentication
- Homepage redirection
- Bandwidth shaping
- Daily time limits

[16] http://www.knoppix.org/
[17] http://httpd.apache.org/
[18] http://www.squid-cache.org/
[19] http://dansguardian.org/

- Daily download limits
- Zone open and close times
- Block by MAC address
- End-user permissions – user classes
- Customized firewall rules per user class
- Content filtering
- Daily log mailing program
- Customize ZoneCD login pages
- Usage statistics
- End-user reporting

Some of the advantages of the ZoneCD implementation are:

- The ZoneCD is distributed on the GPL license which means it's available to the business to use and to modify as per their specific needs.
- The cost of acquisition of the system is minimum since it's a free software.
- ZoneCD is a live CD distribution, which means it runs from the CD without needing to be installed on the hard drive. This is an advantage as it entails the business can run the ZoneCD without altering the software setup on their systems.
- It's a modularized implementation as far as the different components that achieve the total WIFI provisioning functionality are concerned. The advantage is that the individual components of the systems are continually maintained and improved e.g. a newer version of the Apache server, or Squid proxy server, which can be included in the distribution to provide added functionality.

And the disadvantages associated with ZoneCD are:

- The level of technical expertise required may be high for non-standard implementations of the system.
- Since the system runs as a live CD, it is recommended that regular system auto-restarts be schedule in order to alleviate issues regarding memory allocation and resources overload.
- The perceived ease of use of the system might be low due to the fact that it's a Linux based system.

6.2.3 NoCatAuth[20]

NoCatAuth is an open source Linux-based authentication system that can be implemented in a variety of wireless service deployments. The system provides a centralized authentication infrastructure that comprises the gateway server, the authentication server and an access point (Figure 6-6). The function of the gateway server is to handle incoming client connection requests, issue IP address, enforce access permission to resources and handle bandwidth throttling. The authentication server on the other hand is responsible for looking up user's credentials in a MySQL database and notifying the gateway server of the users' status. NaCatAuth provides a functionality of capturing the unauthenticated users' requests and redirecting them to an authentication portal.

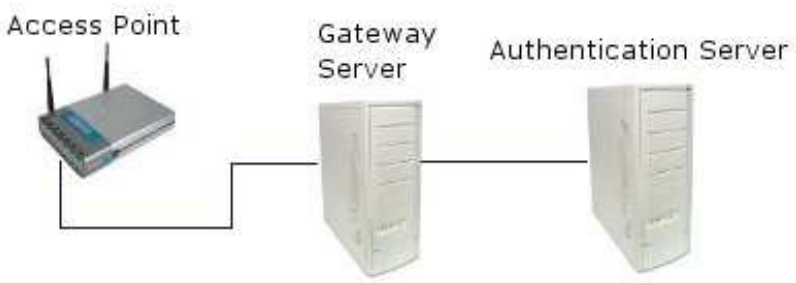

FIGURE 6-6 NOCATAUTH SETUP

The connection process via the NaCatAuth server is as follows (Nocat, 2005):

- Redirect - a user associates with an AP and is issues an IP via DHCP. At this point all access is denied except to the Auth service so when a user attempts to browse the internet, the system redirects them to an SSL authentication portal.
- Connect Back - after providing the login credentials, the Auth system creates a PGP signed message of the outcome of the authentication attempt and sends it to the gateway.
- Pass Through - if the user provided accurate credentials, the gateway modifies its firewall rules to grand further access to the user.

[20] Downloadable from http://nocat.net/download/

6.2.3.1 NoCatAuth vs. SEHS Authentication

The authentication infrastructure implemented in SEHS is different from that provided by NoCatAuth in a sense that it is not a third party server based authentication. In SEHS, the authentication process is implemented in a `serviceElement` module (i.e., `x_se_httpGateKeeper`). This allows for authentication to be service-specific since some services might not need user authentication on the system (e.g., `x_se_http` which provides free open internet access). The different authentication related functionality and how it's implemented both in NoCatAuth™ and SEHS is as follows.

Functionality	NoCatAuth™	SEHS
Redirecting users	Gateway service	`x_se_httpGateKeeper` (SE)
Authenticating users	Auth service	`x_se_auth` (SE)
Connection management	Gateway service	`x_adaptor`
Access permissions	Gateway service	`x_se_httpGateKeeper` (SE)
DHCP	Gateway service	AP

Connection authentication process via SEHS is as follows

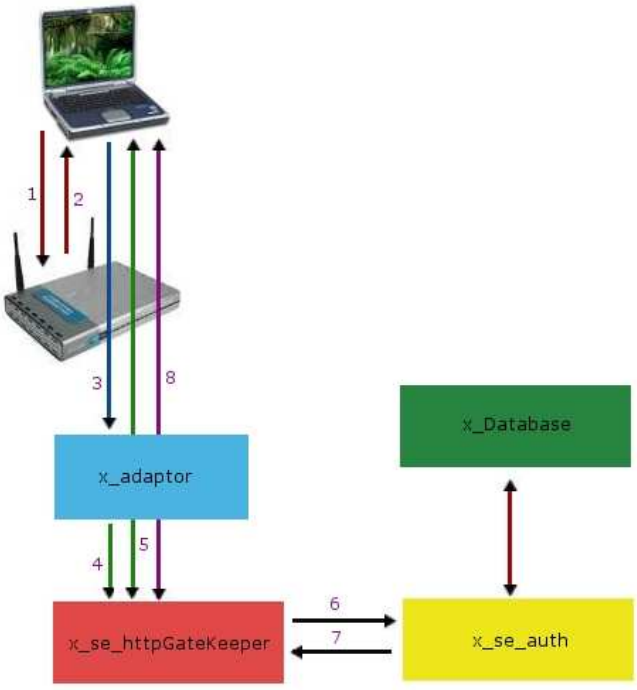

FIGURE 6-7 AUTHENTICATION PROCESS IN SEHS

The steps in the authentication process in SEHS, depicted in Figure 6-7, are as follows:

1. DHCPREQUEST message: When a client associates with an AP it sends a DHCPDISCOVER message to determine the availability of a DHCP server. The AP, which provides the DHCP functionality, then responds with a DHCPOFFER message specifying the configuration parameters. Subsequent to the offer, the client then sends the DHCPREQUEST message.
2. DHCPACK message: When the AP receives the DHCPREQUEST message, it constructs a DHCPACK to acknowledge and confirm the offer. Other messages that can be sent are DHCPNOACK to reject the offer and DHCPDECLINE to indicated that the address is in use already.
3. Connection to SEHS server port: Once the network layer configuration is done, a client connects to the `x_adaptor` through an open SESH port that is configured via a client's proxy settings. At this stage, the client is given access to the system in terms of requesting services via the `x_adaptor`. Depending on the authentication requirements of the executing service element, further authentication would be done.
4. Service Request to `x_se_httpGateKeeper`: before providing the service to the user, `x_se_httpGateKeeper` checks their authentication status. The different statuses that are defined for a user connection are discussed in section 5.5.1.
5. Redirect to authentication page: The user enters their login credentials and submits the login request to the `x_se_httpGateKeeper`.
6. Authenticate via `x_se_auth`: The login credentials are forwarded to the `x_se_auth` for authentication.
7. Authenticate request status: `x_se_auth` responds with the status of the authentication request i.e. whether the client provided correct credentials or not.
8. Service execution: on successful authentication, the user is then able to access the service that is provided by the `x_se_httpGateKeeper` SE.

The authentication infrastructure that is provided by NoCatAuth is provided on the Linux platform because it is based on the *iptables* functionality that is implemented in

the Linux operating system. As indicated in section 3.1.3.1 Linux is not the prominently used operating system among the small scale businesses that were profiled and hence the limitation is introduced in terms of usability and applicability for small PALs who may not prepared to install the Linux operating system. The SEHS architecture on the other hand is designed to be implemented across different platforms.

The authentication infrastructure implemented in SEHS is provided via service element modules. This is motivated by the extensibility design goal of the system. This infrastructure allows for different authentication method and protocols to be implemented in the service element modules as opposed to being hard coded into the gateway component of the system. In the current implementation of SEHS, `x_se_auth` provides an authentication against a MySQL database, hence the reason why the authentication process is executed via the `x_Database` component (Figure 6-7). It's also a more flexile architecture as it allows for a pluggable infrastructure for the authentication service element modules. The clients are able to write their own authentication modules and to then plug them into the system to extend the authentication functionality or to use third party authentication modules that provide predefined authentication operations (e.g., RADIUS authentication, central server authentication).

6.3 Applicability of IPDR within the context of small WISPs

The IPDR specification provides an interface that is used by IP networks and service providers to capture network data usage information (IPDR, 2005). The classical implementation of mediation solutions necessitated a development of new adapters for every service element module due to the different usage accounting protocols that are implemented. The IPDR specification provides a usage accounting format in the form of Network Data Management-Usage (NDM-U) specification which alleviates the problems associated with providing a mediation solution for different service element modules and also the problem associated with the exchange of usage data between different business units.

The NDM-U specification is an XML based usage format that allows for the extensibility of the network usage record. The IPDR specification also defines a communication protocol for a secure, efficient transportation of usage information between the business units. Due to the extensible nature of the specification, it can support usage accounting for a large variety of network services (e.g., VOIP, streaming video, WLAN).

The IPDR specification provides an accounting and mediation solution that is designed to cater for current and next-generation IP network services. It is a solution that spreads through a large spectrum of network services provisioning supply chain, from the service providers, mediation systems vendors, billing system vendor to carrier network operators, clearance houses, etc.

The advantages associated with the IPDR standard are the following (IPDR, 2005):

- The specification allows for an implementation of systems that support evolving business requirements.
- It provides support for real time and high performance requirements of next generation data networks

Other advantages of the IPDR specification are provided below juxtaposed with other usage accounting solutions:

- *IPDR Versus RADIUS* - Radius has limited extensibility features as opposed to IPDR which implements a usage format that is inherently extensible.
- *IPDR Versus SNMP* - SNMP requires lot of processing and bandwidth in order to handle the usage data. IPDR on the other hand depends on a defined streaming protocol for the exchange of usage information and a simple text based encoding standard for the actual data (i.e. XML).
- *IPDR Versus DIAMETER* - Diameter is inadequate in addressing accounting protocol issues while IPDR is specifically geared towards solving accounting related issues and for handling different service-specific accounting protocols.
- *IPDR Versus CDR* - CDR is based on one specific service usage accounting. The data record is not extensible to be implemented for a different service.

The functionality that is beneficial to small hotspot management system implementation that has been realised through the usage of the IPDR specification includes:

- Usage accounting - At the core of the IPDR standard is a mechanism that allows for network usage accounting. Small hotspots need the accounting functionality in order to be able to perform business related operations based on the usage of the system, e.g. charge for usage, analysis of the accounting data to identify usage patterns of the users. Figure 6-8 is an example of a usage record on the SEHS using the IPDRDoc format. It shows the usage of the SEHS on the `X_HTTP_COUPONKEEPER` service element.

```
<?xml version="1.0" ?>
- <IPDRDoc xmlns="http://www.ipdr.org./namespaces/ipdr" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  instance" xsi:schemaLocation="http://www.ipdr.org./namespaces/ipdr x_http_couponkeeper.xsd"
  docId="2f401524-0105-5000-eac9-d933c0a80064" creationTime="2005-07-19T13:19:53.124Z"
  IPDRRecorderInfo="IPDR-Xobogel" version="3.1">
- <IPDR xsi:type="X_HTTP_COUPONKEEPER">
    <IPDRCreationTime>2005-07-19T13:19:53.124Z</IPDRCreationTime>
    <seqNum>1</seqNum>
    <clientID>userthree</clientID>
    <clientIP>146.231.121.160</clientIP>
    <startTime>792536258</startTime>
    <endTime>792710604</endTime>
    <usageUnit>dataBytes</usageUnit>
    <quantity>105719</quantity>
    <requestCount>32</requestCount>
  </IPDR>
- <IPDR xsi:type="X_HTTP_COUPONKEEPER">
    <IPDRCreationTime>2005-07-19T13:19:53.124Z</IPDRCreationTime>
    <seqNum>2</seqNum>
    <clientID>userone</clientID>
    <clientIP>192.168.0.101</clientIP>
    <startTime>792480725</startTime>
    <endTime>792728853</endTime>
    <usageUnit>dataBytes</usageUnit>
    <quantity>54471</quantity>
    <requestCount>16</requestCount>
  </IPDR>
```

FIGURE 6-8 USAGE ACCOUNTING ON SEHS

- Implementation of an extensible, flexible billing framework - The IPDR standard has facilitated the implementation of a billing framework that is generic and flexible to handle different billing modules and different service element modules. The standard has also facilitated the provisioning of a solution to the usage metrics ambiguity problem discussed in section 5.5.2.
- Usage accounting for diverse service elements - The IPDRDoc format specification is based on XML and as a result is extensible to accommodate the different usage metrics associated with diverse service elements. As discussed in section 4.8, implementing accounting measures for a new service

element is just a matter of defining a new schema definition file for the specific module and plugging it into the system.

- Access to third party modules - The IPDR specification provides a standard interface that allows interoperability with third party accounting and mediation packages. Since SEHS implements the IPDR standard, it means that other billing modules that are IPDR compliant can be utilized on the system and other mediation systems can be integrated into SEHS.
- Industry standard solutions - The IPDR specification has been developed in collaboration with large, well established companies in the telecommunication service accounting industry (IPDR, 2005). Implementing IPDR in the context of SEHS (small PAL system) allows for industry standard solutions and expertise to be implemented on a smaller scale and for the associated benefits to be realised by small PALs.

6.4 Efficacy of the microkernel architectural pattern

The central building block of the Xobogel framework and hence the architectural basis of SEHS is the microkernel architectural pattern. The pattern is introduced in section 4.3 in the context of providing an overview of the Xobogel framework. This section provides the motivation for the choice of this pattern over other architectural patterns that have been developed and subsequently provides an overview of the associated advantages.

Architectural Patterns provide a high level overview of how the individual system components should interact and inter-operate. Instead of providing implementation specific instructions, they provide a very general, abstracted guideline to how the system implementation should be approached. Examples of other architectural frameworks and the context within which they are applied are shown in Table 6-4.

TABLE 6-4 ARCHITECTURAL PATTERNS

SYSTEM FUNCTIONALITY AND CONTEXT	PATTERN
Distributed system	Broker
Interactive Systems	Model-View-Controller

	Presentation-Abstraction-Control
High level decompositions	Layers
	Pipes and filters
	Blackboard
Adaptable Systems	Microkernel
	Reflection

The microkernel architecture provides an implementation approach for systems that must adapt to changing requirements and environments. Extensibility and adaptability are some of the most essential features that need to be provided in a system that address the requirements for PALs. This is due to the following reasons:

- The computing industry (hence networking) is a very dynamic industry where changes and improvements are always happening i.e. new protocols, new standards, operation improvements etc, and as a result the systems implemented for wireless network service provisioning, must be able to adapt to the rapidly changing environmental (industry) factors.
- The users' requirements for PAL systems differ depending on the specific environment within which the system is implemented. In order to cater for these specific requirements, the system must be adaptable and extensible.

The advantages that are realised through the use of the microkernel architecture for the PAL management system are as follows:

- Porting to a new environment does not need changes to the external servers (i.e., changing the operating platform for the Xobogel based system does not necessitate changing any functionality in the service element modules).
- External servers can be maintained independently of the kernel. In the case of Xobogel; the service element modules can be maintained independently of the kernel and it provides a mechanism for third party service element modules to be used in the system.
- Extending the functionality of the system is a matter of adding new external servers and in the case of Xobogel it is a matter and defining and adding new service element modules.

6.4.1 Microkernel vs. Monolithic architecture

The microkernel architecture is a more flexible architecture than the monolithic architectures. This is because monolithic architectures provide for a fixed limited functionality that is embedded into the system. Extending the functionality of a monolithic architecture-based system requires the system to be shut down, the new functionality added in the system, the whole system recompiled and then restarted to allow the new functionality to take effect. This approach is cumbersome and can be expensive to undertake.

6.4.2 Microkernel vs. Reflection pattern

The other architectural pattern for adaptable system that could have been implemented is the Reflection pattern. The basic building block of the reflection pattern is to allow the system to be able to provide information about itself. This is done via meta-level information about the system and by implementing the ability for the system to modify its behaviour. The Reflection pattern is defined formally as:

> *"The Reflection architectural pattern provides a mechanism for changing structure and behaviour of software systems dynamically. It supports the modification of fundamental aspects, such as type structure and function call mechanisms. In this pattern, an application is split into two parts. A meta level provides information about selected system properties and makes the software self-aware. A base level includes the application logic. Its implementation builds on the meta level. Changes to information kept in the meta level affect subsequent base level behaviour."* (Suzuki et al, 1999)

The meta level allows for a definition of base objects which are specific to the requirements within which the system in being implemented. It is at the meta-level that the functionality of the system is changed to allow for extensibility and adaptability. The application level objects subsequently provide the functionality that is defined in the meta-level base objects. The Reflection pattern's advantage is that it provides for a decoupled definition of the meta-level and the application level base objects thus allowing for a transparent independent evolution of both the meta level and the application level (Suzuki et al, 1999).

The reflection object introduces an increased complexity to the system since the meta level definition of bases objects increases the number of application level objects that have to be defined. Executing an operation on the Reflection pattern based system requires communication between the meta level and the application level objects and thus it is slow and inefficient as far as task completion is concerned. The efficient running of the system is dependent on the robustness of the meta level objects which may render the system not operational if not properly modified.

The micro-kernel on the other hand provides adaptability through a plug and play mechanism that is implemented in the central core of the system (i.e., the kernel). The functionality of the microkernel system is extended via the external servers which are autonomous self contained modules. The external servers only need to be aware of the interaction interface with the kernel in order to plug into the system. This is the desired functionality since the external server functionality in Xobogel is provided via service element modules which need to be self contained. The service element should encapsulate the total functionality for the service that they provide (i.e., the http service element should provide the functionality needed for the http service provisioning). The basis of interaction between the service element modules and the kernel is the predefined interface that both the kernel and the service element are aware of. This architecture minimises the need for the high level meta information about the system which is implemented in the Reflection pattern. The microkernel architectural pattern is best suited for the purpose of meeting the identified requirements for PALs as it provides an approach to system design that is adaptable, modularised, extensible and easy to implement.

6.5 One thread per client vs. selector model

The classical implementation of socket programming, pre java v1.4, involved creating a new thread for every new client connection to the server socket (Listing 6-1). This multithreaded model was, and currently still is, used to implement server applications in java. In this model, a new path of execution is created for the new connections while still sharing the original data area of the parent thread. This model is less CPU intensive than an explicit `fork()` statement which creates a different address space both for code and for data.

```
    public void go()
    {
     try
       {listen = new ServerSocket(port);
        while(true)
          {new handleCon(listen.accept()).start();
          }
       }
      catch(Exception e)
       {System.err.println(e.getMessage());
       }
}


 private class handleCon extends Thread
  {Socket socket = null;
   InputStream in = null;
   OutputStream out = null;

   public handleCon(Socket a)
   {socket = a;
   }
```

LISTING 6-1 ONE THREAD PER CLIENT MODEL

Java introduced, with the release of Java v1.4, the Java New I/O (JNIO) API that allows for a new model of handling multiple connections. This model uses selectors to manage multiple connections executing on a single thread. A selector is an object that maintains and monitors the recorded socket channels and then serializes the requests for the server to handle. The function of the selector object is to provide a built-in multiplexed I/O functionality. This is done by registering a socket channel with a selector object specifying the task that the channel will perform in the program.

Table 6-5 shows the different tasks that a socket channel can perform and Listing 6-2 shows an example of registering a channel that both reads and writes.

TABLE 6-5 SELECTIONKEY TASKS

Field Summary	
static int	**OP_ACCEPT** Operation-set bit for socket-accept operations.
static int	**OP_CONNECT** Operation-set bit for socket-connect operations.
static int	**OP_READ** Operation-set bit for read operations.
static int	**OP_WRITE** Operation-set bit for write operations.

```
SelectionKey readWriteKey = channel.register(selector, SelectionKey.OP_READ | SelectionKey.OP_WRITE);
```

LISTING 6-2 REGISTERING A READ AND WRITE SOCKET CHANNEL WITH A SELECTOR

When a channel is registered with a selector, a `SelectionKey` is generated to which the `socketchannel` is attached for further usage. The `SelectionKeys` have boolean flags associated with the task that the underlying socket channel performs;

- *key.isAcceptable()* : for a channel that performs the OP_CONNECT.
- *key.isWritable()* : for an OP_WRITE channel.
- *key.isReadable()* : for a channel that performs OP_READ.
- *key.isConnectable()* : for a channel that has the OP_CONNECT task.

The main loop of the application simply queries the selector object for available keys. As the keys are returned from the selector object, the program queries the boolean flag of the key to see what type of a key it is and then performs the required function for that task. For example, if the key is readable, the program simply queries the key for the underlying socket channel and then reads from the associated input stream, if the key is writable, the program writes some data to the associated output stream.

In an experiment that was undertaken on the performance of JNIO vs. the thread per client model (Cowan T., 2004). It was found that the JNIO architecture scales better than standard IO and that JNIO handles many clients with a fixed number of threads.

This experiment was performed on a Tomcat 5.0 server and a JNIO server. Tomcat is a 100 percent java solution that implements the standard Java IO API.

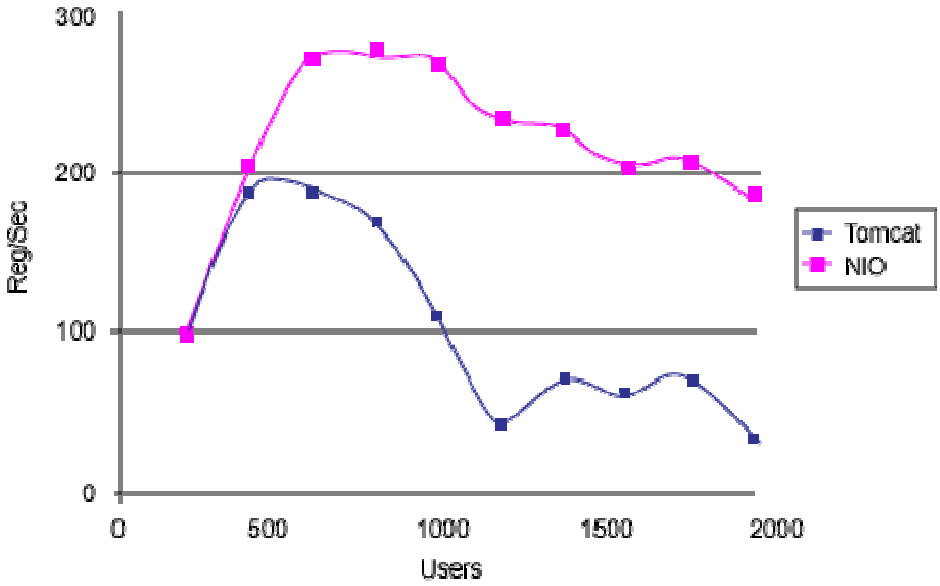

FIGURE 6-9 JNIO SERVER VS. TOMCAT (COWAN T., 2004)

The two architectures show similar, comparable performance up to a user load of 200 clients (Figure 6-9). Beyond that point (i.e., the 200 clients point) the Tomcat performance begins to deteriorate while the JNIO continues increasing. The JNIO begins decreasing in a linear fashion past the 800 users point.

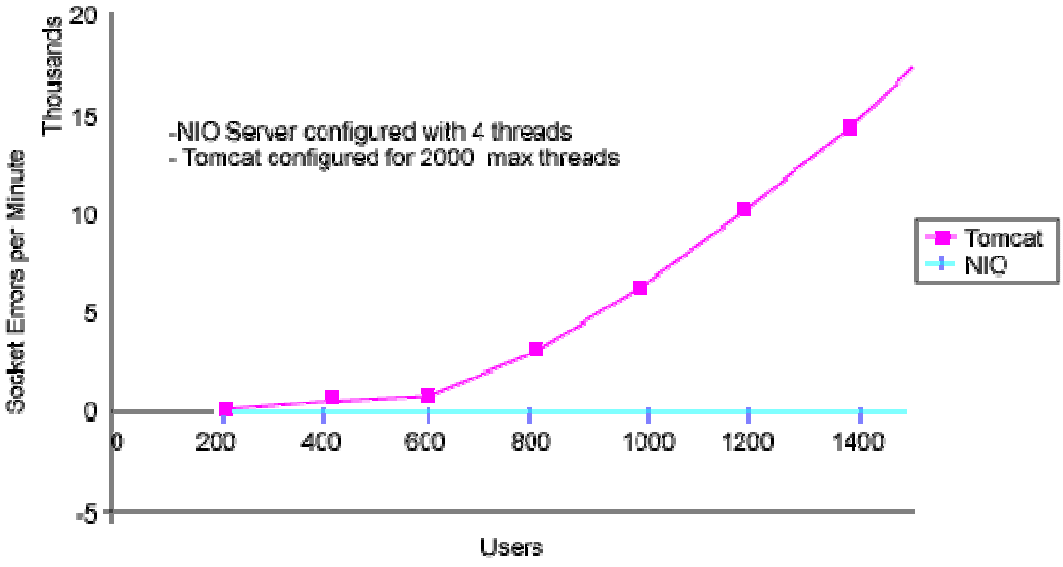

FIGURE 6-10 JNIO SERVER VS. TOMCAT 5.0 SOCKET ERRORS (COWAN T., 2004)

The number of errors generated from the Tomcat server (Figure 6-10) also indicates the lower performance of standard IO versus that of JNIO architecture.

These results indicate the overall performance comparison for the two APIs in the context of large scale server implementations. As indicated in section 3.1 the maximum user density on the PAL is ten and at that point of usage of the APIs, it appears from the graphs that the performance is comparable. In order to validate this observation, an experiment was set up in which two implementations of SEHS were evaluated in terms of performance. The one version of SEHS is the initial implementation that uses standard Java IO API. The other version uses the Java NIO API and is the current implementation of SEHS. In terms of the functionality that these versions have, the earlier version provides a simple http request handling functionality for local resources, while the current version has all the functionality that is sufficient for PAL service provisioning (i.e., billing functionality, extensibility infrastructure based on the Xobogel framework, usage accounting, and authentication).

The experiment entailed a single HTTP GET request for a local resource that is 1MB large. The experiment was set up to simulate one to ten client connections to the server, looping ten times per client. Different metrics were observed and plotted for the standard Java IO server implementation and for the Java NIO server implementation.

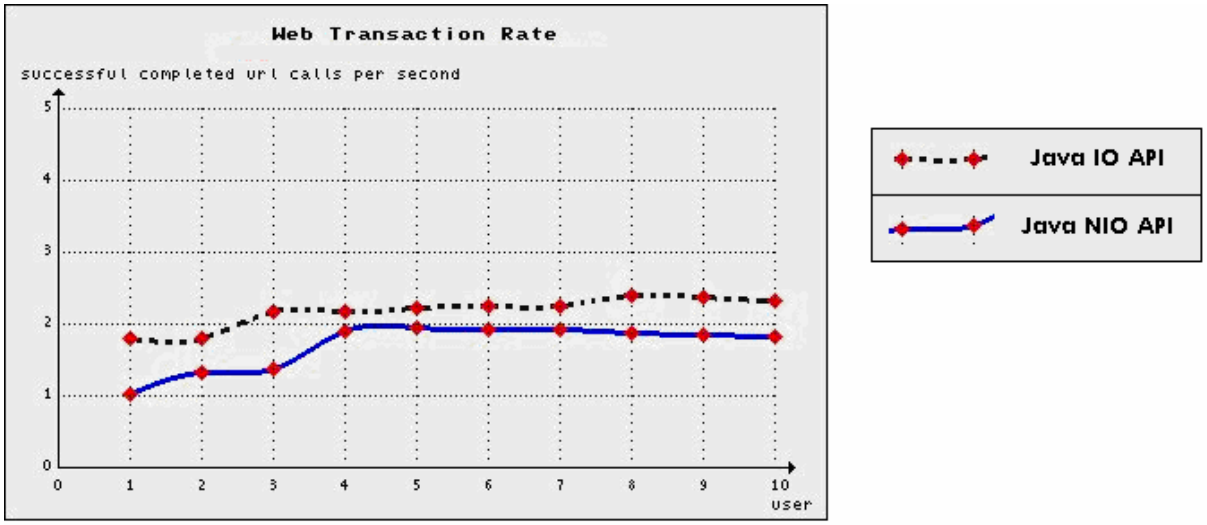

FIGURE 6-11 WEB TRANSACTION RATE

The web transaction rate (Figure 6-11) is a commonly used measurement for web site performance. A web transaction is defined in this case as a single HTTP request and response pair. This metric indicates the total number of completed URL calls that the server was able to handle per second. This metric is measured over al the emulated users.

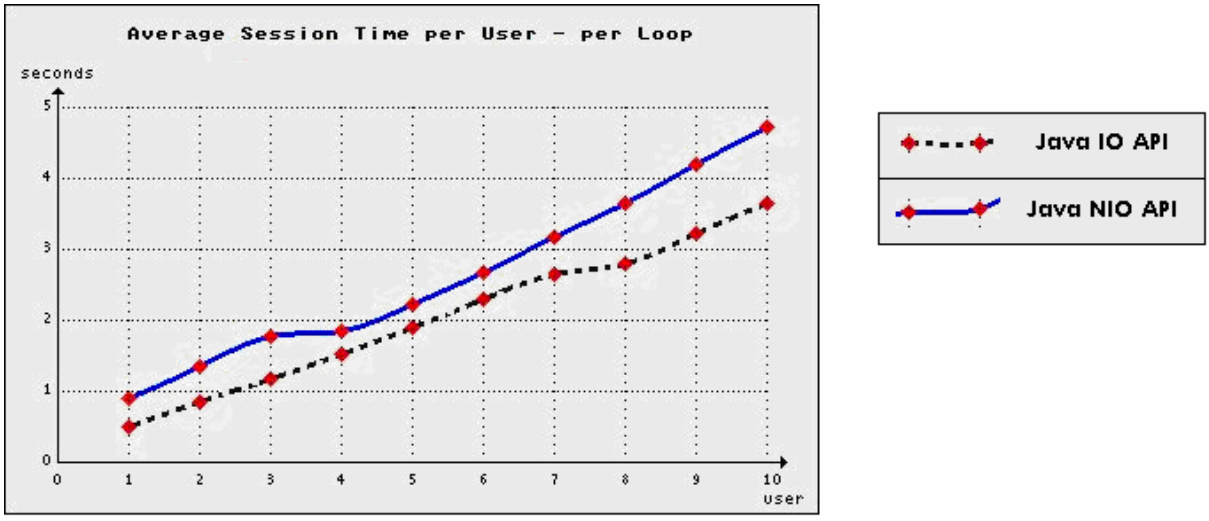

FIGURE 6-12 AVERAGE SESSION TIME PER USER PER LOOP

Average session time metric (Figure 6-12) indicates the average time that a single user spent per web surfing request on the network. This time is measure per each emulated user's usage loop.

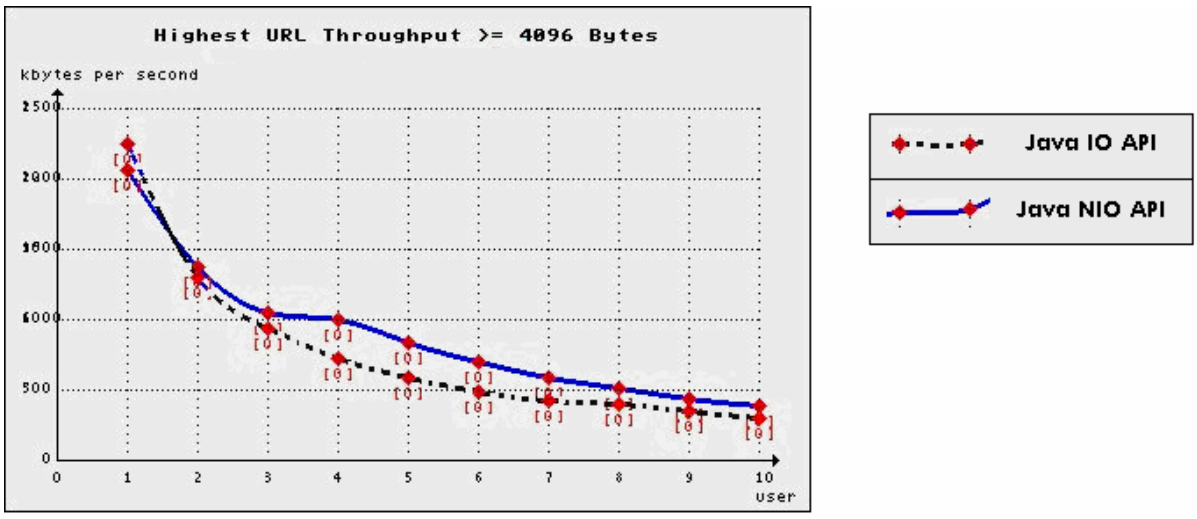

FIGURE 6-13 HIGHEST URL THROUGHPUT

This highest URL throughput metric (Figure 6-13) indicates the average number of kilobytes per second of the fastest URL which has exchanged at least 4096 bytes of data with the server. The index of the fasted URL is indicated in the red brackets.

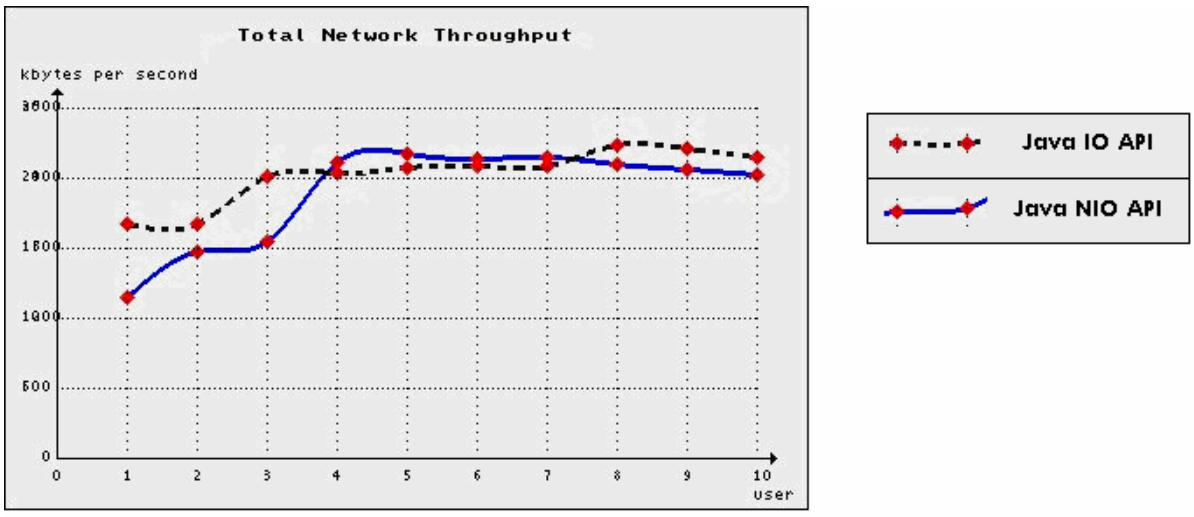

FIGURE 6-14 TOTAL NETWORK THROUGHPUT

The total network throughput metric displays the total number of kilobytes transferred per second across all the users.

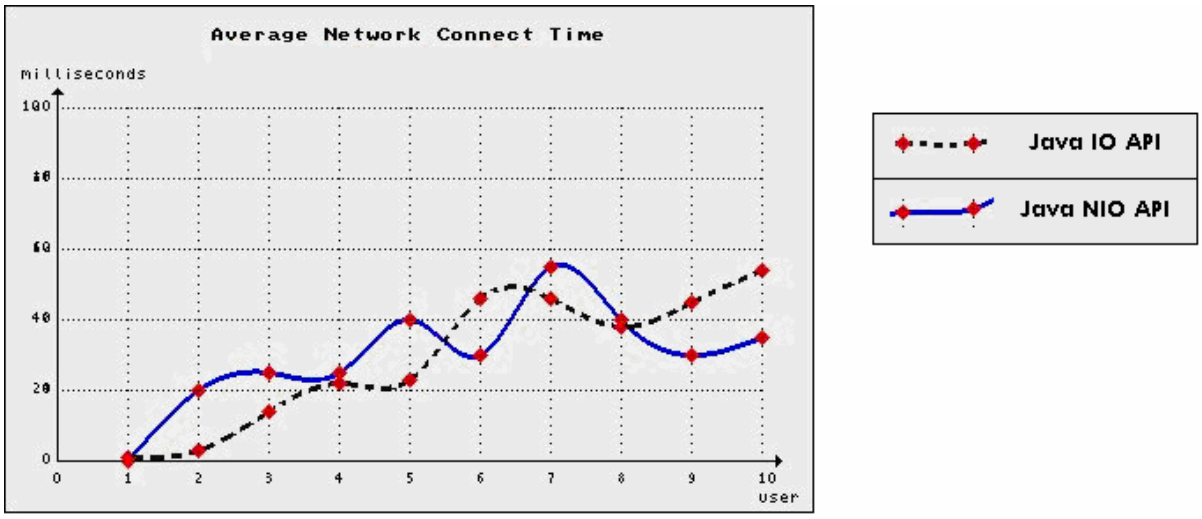

FIGURE 6-15 AVERAGE NETWORK CONNECT TIME

The average network connect time metric (Figure 6-15) shows the average time per URL call to establish a network connection to the server before the data is exchanged.

Figure 6-11 to Figure 6-15 indicate the results from the experiment. The solid line in the graphs represents the results observed on the JNIO API server implementation and the dotted line represents the standard Java IO server implementation. The following observation can be made from the results:

- At low usage levels, the performance of standard IO is comparable to that of JNIO API.
- A slightly better performance is observed for the standard Java IO in the above experiment than for JNIO (i.e., an overall higher web transaction rate and a lower average session time per user per loop). A number of reasons suffice to explain this observation:
    - The JNIO server architecture is more complex than that of the standard IO server. The Xobogel architectural framework which is implemented in the JNIO server is based on the microkernel architecture which has a more complex request handling mechanisms.

- More functionality is implemented in the JNIO server than in the standard Java IO server.
- The HTTP handling functionality in the JNIO server is handled in a SE module which increases the function call time associated with the JNIO server.

While the performance of the two APIs seem comparable at the usage levels at which the system is implemented, the JNIO is a better API at much higher usage levels and also it eliminates the issues associated with the thread per client model of pre Java v1.4 (e.g., deadlocks, thread safety violations, concurrency, and conflict resolution). Another reason why the JNIO API is more preferable is due to its non-blocking IO capabilities.

6.6 Synchronous vs. Asynchronous I/O

Synchronous IO is the method of doing IO operations in which program execution halts until the IO operation is completed. While the IO operation is executing, the running thread is not able to execute anything else on the system. This is a large amount of time as far as CPU time is concerned. Synchronous IO is the IO implementation that is provided in the pre-version 1.4 standard Java API. Asynchronous IO on the other hand allows for a mechanism where program execution does not halt for the IO operation to complete. When the IO operation is called, the program only has the ability to determine how much data has be send or received or whether the IO operation is complete. Asynchronous IO does not entail that the actual IO process is faster than in synchronous IO rather that the program is able to execute some other section of the code while IO is being done.

The asynchronous IO API forms the basis of the network IO operations in SEHS and as thus allows the system to leverage on the available benefits associated with the API. These are in terms of efficient utilization of system resources, fast execution (in terms of available CPU time to execute other code) and an improved IO infrastructure. SEHS therefore provides a more efficient usage of processing resources.

6.7 Simplicity and Extensibility

The two RECOMMENDED requirements for small PALs solutions are simplicity and extensibility (section 2.1). The extensibility requirement has been met in both the Xobogel framework and the SEHS system in the following ways:

- The Xobogel framework is built upon an extensible microkernel architectural framework (section 4.3).
- The extensibility of the SEHS system has been validated by the implementation of various service element extension modules (section 5.4 and Appendix B: Example service elements).
- Extensibility is also achieved in the billing infrastructure of Xobogel and this has been validated by the implementation of different billing modules in SEHS (section 5.5.2).
- The IPDR specification implemented in the system facilitates and enables the usage accounting for various services provided on the system.

Simplicity can be articulated both in terms of the overall system design and in terms of the ease of use of the system. In the Xobogel framework, simplicity and has been balanced out with the provisioning of the extensibility mechanisms in the framework. In providing extensibility hooks into the Xobogel framework, the microkernel architecture was utilized, which has a relatively complicated message passing mechanism between the components of the architecture (i.e., communication between the client device and the service is not direct, rather it goes through an adaptor, an external server, an engine and then ultimately a service element module). The notion of simplicity that is important to consider however is with regards to the user's experience of utilizing the system. The following can be noted in this regard:

- The administration of SEHS is facilitated by a simple GUI front-end (Figure 6-16), which exposes the basic functionality (e.g., managing service element modules, managing users, managing billing modules, and viewing system logs) needed to run the system. This simplifies the use of the system and adds to the usability of the system.

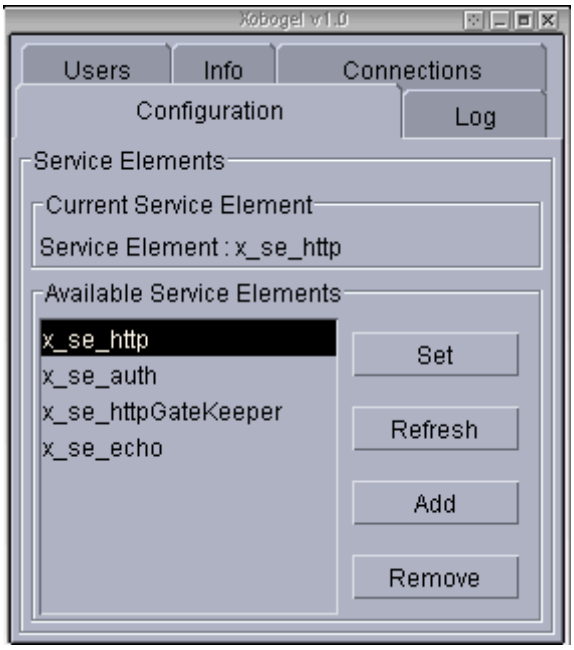

FIGURE 6-16 SEHS FRONT END

- Since an elaborate usability test has not been performed on the system (because it is beyond the scope of the project), a validation of the simplicity of the system is based on the inherent simplicity integrated in the overall design of both Xobogel (section 4.1) and SEHS (e.g., modularity, flexibility).

6.8 Chapter Summary

The SEHS system has been tested and validated to meet the performance requirements that were highlighted in chapter 3. This was done through load testing the system and observing its performance under increased user loads. The unique characteristics and features of the SEHS have also been highlighted in juxtaposition to the other currently available hotspot management systems; these include the flexibility and extensibility of the system which is facilitated by the IPDR specification. Measures that are implemented in the system to meet the overall requirements of extensibility and simplicity are discussed and showed to be sufficient for the purposes highlighted in chapter 3.

7 Chapter 7: Conclusion

This research has been prompted by the current proliferation of PALs, which is increasing the levels of ubiquitous computing. The focus of the research was specifically on small PALs in terms of deriving an architectural framework and implementing, on a proof-of-concept scale, a prototype that implements the derived framework. In view of the need to implement solutions for small PALs, a comprehensive requirements elicitation was undertaken both for PALs users and the small businesses that are potential PAL service providers. The requirements that we determined formed the basis for the Xobogel framework. Subsequent to the derivation of Xobogel (discussed in chapter 4), SEHS was implemented to concretize and to validate the applicability of the Xobogel framework in terms of providing an architectural framework that is sufficient for meeting the requirements for small PALs.

7.1 Xobogel: the applicability of the architectural framework

The design of the Xobogel framework is built upon the microkernel architecture which provides inherent extensibility features via the role of the kernel and the external servers (Section 4.3.2). It is also built on top of the IPDR standard which provides the functionality needed for AAA operations in terms of providing an extensible usage accounting mechanism. The sufficiency of this architecture is in the fact that it is extensible and adaptable to different implementation environments, and to different user requirements. The framework itself is non-specific in terms of the implementation variables (i.e., it is not tied to a specific platform, or implementation language) and hence in this particular case it was implemented on the MS Windows OS using JAVA. This flexibility in the framework makes it applicable in a plethora of scenarios and environments. The flexibility is not only in terms of the implementation of the framework but also in terms of the operation of the implemented system. This is achieved via an extensibility mechanism which allows different service elements to be plugged onto the system in order to achieve the required overall functionality.

The applicability of the framework has been validated by the successful implementation of SEHS which is based entirely on the specification of Xobogel. A

number of different service element modules have been developed (Appendix B: Example service elements) and implemented to confirm the extensibility and flexibility of the framework. The adequacy of the IPDR standard in providing a flexible usage accounting mechanisms has also been validated via the implementation of billing infrastructure in SEHS, detailed in section 5.5.2.

7.2 Efficacy of SEHS for small PALs

The implementation of SEHS has been to validate the Xobogel framework in terms of its applicability in providing solutions for small PALs. As discussed in section 4.1, the design of the Xobogel framework has been guided by the requirements that were ascertained both for users and small WISPs. SEHS is a working PAL management system that can handle basic provisioning of public internet access. The features that have been provided in the system include:

- An open mode via a service element module (`x_se_http`) which allows every device access to the http service.
- A closed mode in which only authentication users are allowed access to the http service. This is also provided via a service element module (`x_se_httpGateKeeper`).
- Basic User management features that allow for adding new users, deleting user accounts and checking users' balances.
- A simple system usage log that keeps an audit of client connections and service element module changes on the system.
- A billing infrastructure that allows for plugging in of different billing modules and their configuration based on the preferences of the WISP. The system also allows for a mechanism to by-pass the billing system in cases where WISPs prefer a free access business model.
- A prepaid mechanism, implemented via a coupon system which allows WISPs to generate time or data based coupons (section 5.5.3) for users to utilize on the system.
- A plug-in mechanism implemented both at the levels of the service element modules and the billing modules. This is the central feature of the system as it

allows the necessary (depending on the specific environmental conditions and user requirements) features to be implemented.

The experiments that have been performed on SEHS, as discussed in chapter 6, highlighted the capability of the system to handle PAL service provisioning and also to handle usage loads that are characteristic of small PALs. There are however a few limitations in the system in terms of production grade implementation.

7.2.1 The limitations of SEHS

These are limitations in terms of the plug-in modules that would provide the necessary functionality:

- Security - the amount of security that has been implemented in SEHS is currently minimal. There are no 802.11 security features that are provided by the system (e.g. 802.1x, WPA, WEP). Service element modules (e.g. `x_se_httpGateKeeper`) are currently implementing the necessary security measures for their particular operation (e.g., authentication, encryption). The architecture of the system is such that the necessary security modules can be plugged into the system and handled at the application layer via associated protocols and standards.
- Currently there are no elaborate reporting, control, billing and invoicing functionality implemented in the system. These are the feature that would enhance the usability of the system as far the WISPs are concerned. The billing modules that are currently implemented in the system are flat-rate billing and usage-based billing. The other common billing scheme that would be applicable in this context is Paris Metro Pricing (section 2.8.2.2.3).
- The system is currently implementing a subset of the HTTP1.0 specification. This is a limitation as it means that some of the functionality provided via HTTP1.1 specification would not be available on the PALs (e.g., chunked encoding, connection persistence).
- The system has been implemented on the MS Windows operating system. Porting onto a different platform might necessitate additional setup operations particularly related to IP routing and handling network related functionality. For example, setting up routing between two NICs in Linux would necessitate

configuring iptables or ipchains accordingly, which is different from simply modifying the routing table in MS Windows via the `route` command.

7.2.2 Applicability of SEHS

The Xobogel framework and subsequently the SEHS provide a platform for the proliferation of ubiquitous computing. The development of the framework with a specific focus on small PALs allows for networking solutions that cater for a market segment that has a wide geographical penetration. Small businesses can be found in almost all areas of society, from rural to urban areas, from small cities to large metropolitans. Providing a PAL management solution that caters for this specific group of businesses ensures wider connectivity options and a truly ubiquitous usage experience for the users. The biggest contribution of the framework and the system might be in rural wireless access provisioning, by providing the framework and the tools for the currently existing businesses in these areas to establish PALs. The flexibility that has been integrated into the framework ensures that it can be implemented in different scenarios (e.g., a single proprietorship business in the rural area, a coffee shop in town, a franchise store, or a shopping mall). The easy implementation of Xobogel and deployment of SEHS would also provide an underlying network connectivity infrastructure, specifically in rural areas, for the implementation of IT based projects aimed at community development, bridging the digital divide and establishment of Information Technology Societies (ITS).

7.3 Future research in this area

Possible extensions to the project and further research that can be undertaken are as follows:

- There are current efforts from different stakeholders to bridge the digital divide between the developed countries and developing countries or nationally between urban areas and rural areas. An elaborate study into the efficacy of small PALs in facilitating the bridging of the divide, and in particular through the implementation of Xobogel and SEHS, would be a definite extension to this project (Raven F., 2005).

- Another area of research that can be undertaken might be in terms of how WiMAX will be integrated in to the PALs industry, in the context of last-mile access provisioning for rural deployments in particular. This could be done in conjunction with a study of the applicability of other broadband access technologies in the provisioning of small PALs, and the extent to which the Xobogel framework could be applied in those contexts.

7.4 Overall conclusion

The research undertaken provides a detailed investigation into the requirements of small PALs and also provides a framework that is based on those requirements. The Xobogel framework and the prototype (i.e., SEHS) adequately meet the requirements identified for small PALs in terms of both the users' and the WISPs' expectations. Small scale PALs management systems that are easily deployable and extensible are therefore a feasible solution as has been validated by the implementation of SEHS.

Acronyms

AAA	Authentication, Authorization and Accounting
AES	Advanced Encryption Standard
AP	Access Point
ARP	Address Resolution Protocol
ARPA	Advanced Research Projects Agency
ARPANET	ARPA Network
ATM	Asynchronous Transfer Mode
BARG	Billing Accounting Roaming Group
BRAN	Broadband Radio Access Networks
BS	Base Station
BSS	Basic Service Set
CCK	Complementary Code Keying
CCMP	Counter-Mode CBC-Mac Protocol
CDMA	Code Division Multiple Access
CHAP	Challenge Handshake Authentication Protocol
CIBER	Cellular Inter-carrier Billing Exchange Roamer
CM	Call Management
CPS	Cumulus Pricing Scheme
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DLL	Data Link Layer
DNAT	Dynamic Network Address Translation
DNS	Domain Name Service
DSL	Digital Subscriber Line
DSSS	Direct Sequence Spread Spectrum
EAP	Extensible Authentication Protocol
EAP-TLS	Extensible Authentication Protocol – Transport Layer Security
ECSD	Enhanced Circuit-Switched Data

EDGE	Enhanced Data rates for Global Evolution
EGPRS	Enhance General Packet Radio Service
ESS	Extended Service Set
eTOM	Enhanced Telecommunication Operation Map
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FHSS	Frequency Hoping Spread Spectrum
GPRS	General Packet Radio Service
GRX	GPRS Roaming Exchange
GSM	General System for Mobile communications
GUI	Graphical User Interface
HCI	Host Control Interface
HIPERLAN	High Performance Radio Local Area Network
HIPERMAN	High Performance Radio Metropolitan Area Network
HTTP	Hyper Text Transfer Protocol
IBSS	Independent Basic Service Set
IC	Integrity Check
IEEE	Institute of Electrical and Electronic Engineers
IM	Instant Messaging
IMAP	Internet Mail Access Protocol
IOT	Inter Operator Tariff
IP	Internet Protocol
IPDR	Internet Protocol Data Record
IR	Infra-Red
IRC	Internet Relay Chat
ISDN	Integrated Services Digital Network
ISM	Industrial Scientific and Medical radio bands
ISO	International Standards Organisation
ISP	Internet Service Provider
ISP	Internet Service Provider
IV	Initialization Vector
JVM	JAVA Virtual Machine

L2CAP	Logical Link Control and Adaptation Protocol
LAN	Local Area Networks
LAPD	Link Access Procedure D-Channel
LAWN	Local Area Wireless Network
LEAP	Lightweight Extensible Authentication Protocol
LLC	Logical Link Control
MAC	Medium Access Control
MD4	Message Digest 4
MM	Mobility Management
MS	Mobile Station
MSC	Mobile Switching Center
MS-CHAP	Microsoft Challenge Handshake Authentication Protocol
MTA	Message Transfer Agents
MTP	Message Transfer Part
NAT	Network Address Translation
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
OFDM	Orthogonal Frequency Division Multiplexing
OS	Operating System
OSA	Open System Authentication
OSI	Open System Interconnect
PAL	Public Access Location
PAN	Personal Area Networks
PAP	Password Authentication Protocol
PC	Personal Computer
PDA	Personal Digital Assistants
PDA	Personal Digital Assistants
PHY	Physical Layer
PMP	Paris Metro Pricing
PPP	Point-to-Point Protocol
QoS	Quality of Service
RADIUS	Remote Access Dial In User Service

RF	Radio Frequency
ROI	Return On Investment
RR	Radio Resource
SDP	Service Discovery Protocol
SIM	Subscriber Identity Module
SKA	Shared Key Authentication
SKID	Secret Key Identification Protocol
SLA	Service Level Agreement
SMS	Short Message Service
SPAP	Shiva Password Authentication Protocol
SPAP NT-RAS	SPAP NT Remote Access Server
SS7	Signalling System 7
SSID	Service Set Identifier
TACACS	Terminal Access Controller Access Control System
TADIG	Transferred Account Data Interchange Group
TAP	Transferred Account Procedure
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TKIP	Temporal Key Integrity Protocol
TMF	Telecommunication Management Forum
TTP	Trusted Third Party
UAM	Universal Access Method
UML	Universal Modelling Language
UMTS	Universal Mobile Telecommunications Services
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
URL	Universal Resource Location
UTRAN	UMTS Terrestrial Radio Access Network
UWB	Ultra Wideband
VoIP	Voice over Internet Protocol
VPN	Virtual Private Networks
WAP	Wireless Application Protocol

WEP	Wired Equivalent Privacy
WiBRO	Wireless Broadband
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WISP	Wireless Internet Service Provider
WISP	Wireless Internet Service provider
WLAN	Wireless Local Area Network
WLANAS	WLAN Accounting and Settlement
WORA	Write Once Run Anywhere
WPA	WiFi Protected Access
WPAN	Wireless Personal Area Networks
WPA-PSK	WiFi Protected Access using Pre-Shared Keys
WRAP	Wireless Robust Authentication Protocol
WUSB	Wireless Universal Serial Bus
XML	eXtensible Mark-up Language
XSD	Xml Schema Definition

Appendix A: SEHS PC configuration

The steps to configure routing between 2 NIC in MS Windows XP Professional:

- D:\Documents and Settings\Mamello>route print
  - *To print the current routing table Error! Reference source not found.*

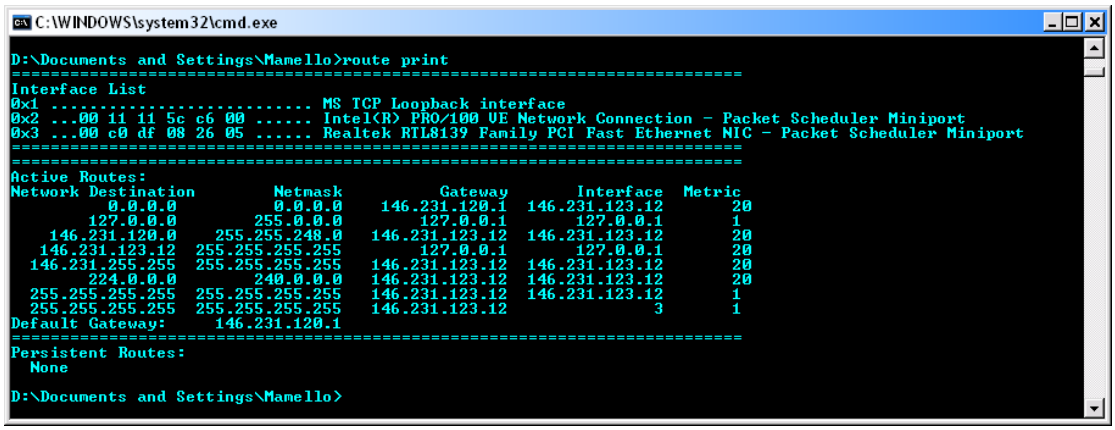

FIGURE A-1 ROUTING TABLE BEFORE 2ND NIC

- D:\Documents and Settings\Mamello>route print
  - *The route table after enabling the 2nd NIC Error! Reference source not found.*

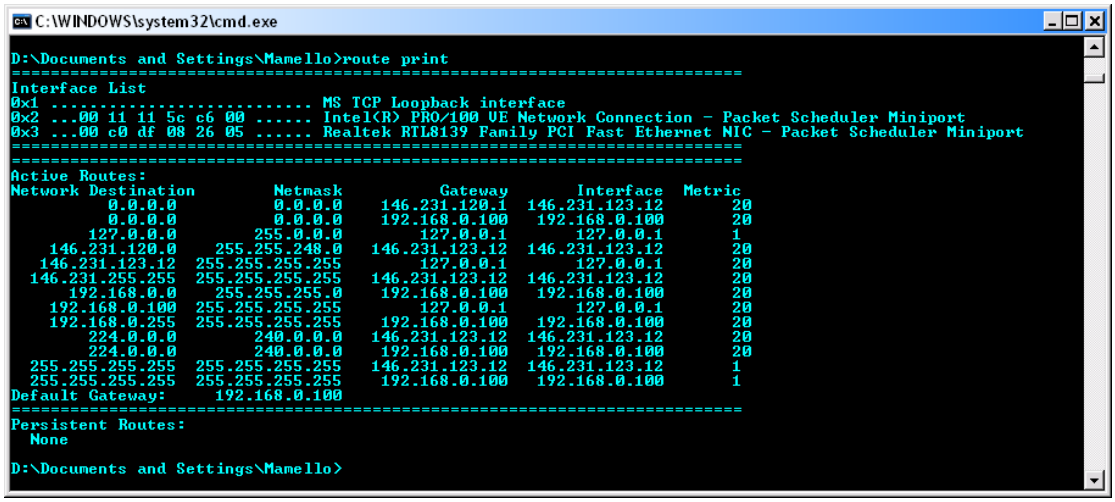

FIGURE A-2 ROUTING TABLE AFTER 2ND NIC

- D:\Documents and Settings\Mamello>route delete 0.0.0.*0*
    - *To delete the 2 conflicting default routes*
- D:\Documents and Settings\Mamello>route add 0.0.0.0 MASK 0.0.0.0 146.231.120.1
    - *To restore the default gateway to the initial LAN interface Error! Reference source not found.*

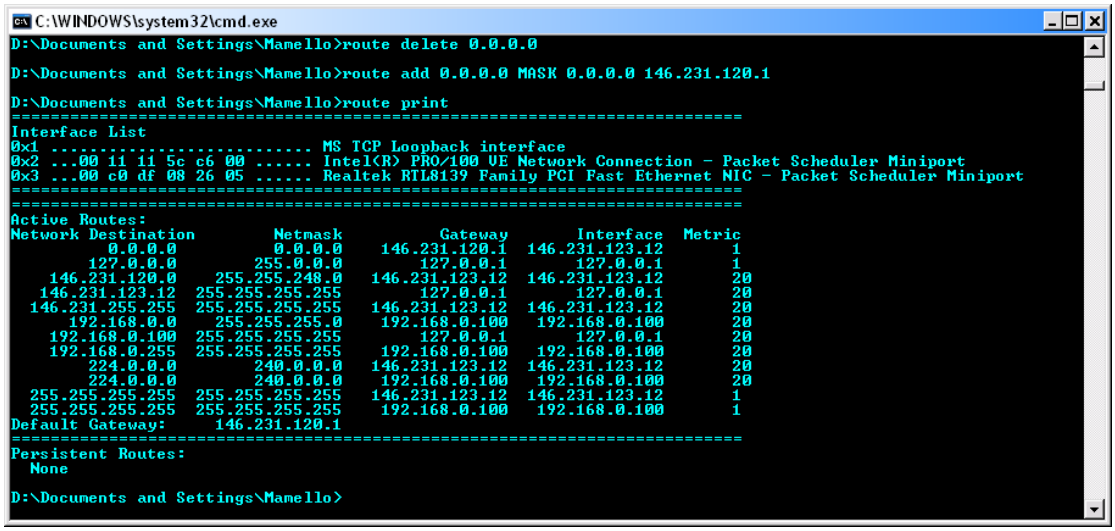

FIGURE A-3 ROUTING TABLE AFTER CONFIGURATION

Appendix B: Example service elements

x_se_httpGateKeeper

- *Description*: this service module extends the x_se_http module to provide AAA enabled HTTP service i.e. authenticated access to the Internet.
- *Usage Scenario*
    - a request (x_req_httpGateKeeper) is received from the x_engine
    - the service module determines if the client has been authenticated
    - If the client device has not been authenticated, it is redirected to the login page
    - If the client device has been authenticated, the service module determines the URI of the requested resource from the request and fetches the resource. It handles both local resources and remote resources, in which case it provides web proxying functionality.
    - a response (x_res_http) is constructed from the information fetched from the internet
    - the response is send back to the request device
    - At the end of the service usage (when user logs out) a usage report is displayed on the client device.

- *Object Diagram*

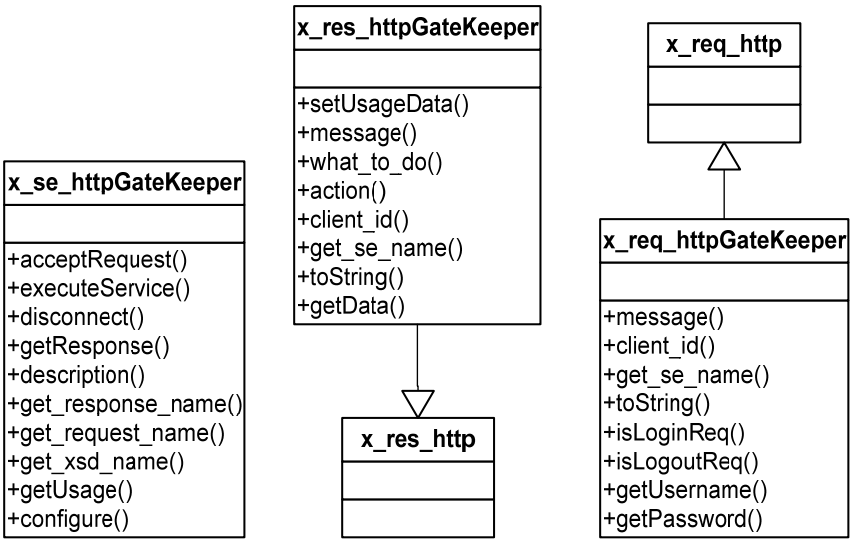

FIGURE A-1 X_SE_HTTPGATEKEEPER DATA OBJECTS

- *Usage metrics*

Table B-1 `x_se_httpGateKeeper` usage metrics

Category	Usage attribute name	Data type	Presence
Who	clientID	String	REQUIRED
Who	clientIP	String	REQUIRED
When	startTime	Date/Time	REQUIRED
When	endTime	Date/Time	REQUIRED
What	usageUnit	String	REQUIRED
What	Quantity	Integer	REQUIRED
What	requestCount	Integer	REQUIRED

- *Usage Schema Definition*

```
<?xml version = "1.0" encoding = "UTF-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
      targetNamespace = "http://www.ipdr.org/namespaces/ipdr"
      xmlns:ipdr = "http://www.ipdr.org/namespaces/ipdr"
      version = "3.0-A.0"
      elementFormDefault = "qualified"
      attributeFormDefault = "unqualified">
     <include schemaLocation = "IPDRDoc3.0.xsd"/>
     <element name = "clientID" type="string"/>
     <element name = "clientIP" type="string"/>
     <element name = "startTime" type = "dateTime"/>
     <element name = "endTime" type = "dateTime"/>
     <element name = "usageUnit">
         <simpleType>
            <restriction base="string">
               <enumeration value="timeSecs"/>
               <enumeration value="timeMins"/>
               <enumeration value="dataBytes"/>
               <enumeration value="dataKbs"/>
            </restriction>
         </simpleType>
     </element>
     <element name = "quantity" type="int"/>
     <element name = "requestCount" type="int"/>
     <complexType name = "X_HTTP_GATEKEEPER">
         <complexContent>
             <extension base = "ipdr:IPDRType">
                 <sequence>
                     <element ref = "ipdr:clientID" />
                     <element ref = "ipdr:clientIP"/>
                     <element ref = "ipdr:startTime"/>
                     <element ref = "ipdr:endTime"/>
                     <element ref = "ipdr:usageUnit"/>
                 </sequence>
             </extension>
         </complexContent>
     </complexType>
</schema>
```

LISTING B-1 X_SE_HTTPGATEKEEPER SCHEMA DEFINITION

- *Screenshots*

FIGURE B-2 X_SE_HTTPGATEKEEPER MODULE SELECTION

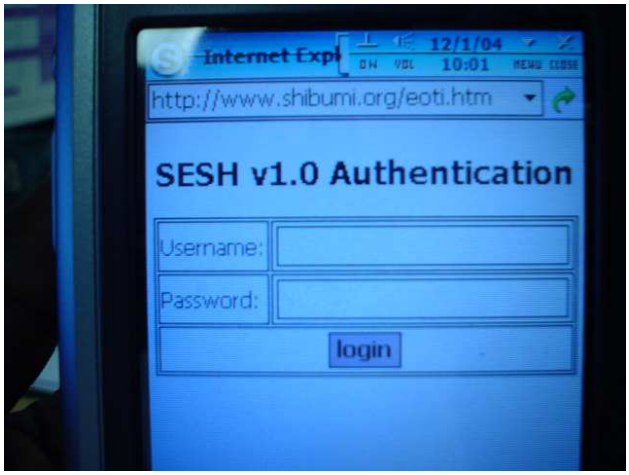

FIGURE B-3 X_SE_HTTPGATEKEEPER LOGIN PAGE

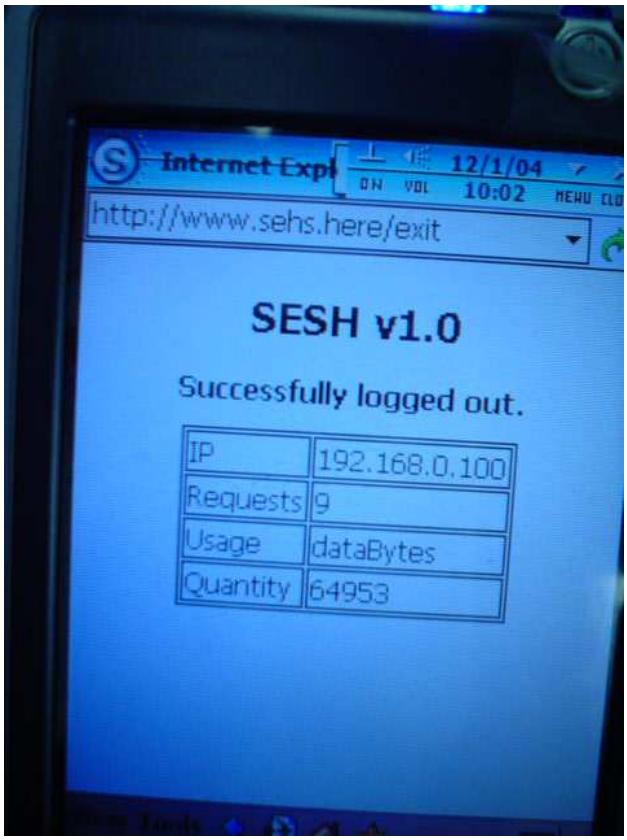

FIGURE B-4 X_SE_HTTPGATEKEEPER END OF SESSION PAGE

x_se_echo

- *Description:* this module simply echoes the request back to the client device.
- *Usage scenario:*
  - A request (`x_req_echo`) is received from the `x_engine`
  - The service module reads the request
  - The request data is send back to the client device.
- *Object Diagram*

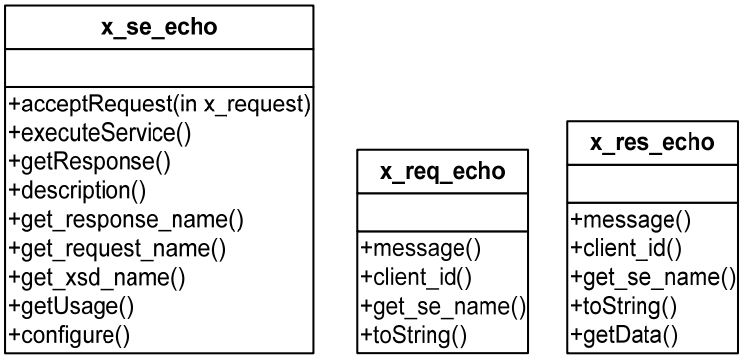

FIGURE B-5 X_SE_ECHO DATA OBJECTS

- *Usage metrics*

Table B-2 x_se_echo usage metrics

Category	Usage attribute name	Data type	Presence
Who	clientIP	String	REQUIRED
When	startTime	Date/Time	REQUIRED

- *Usage schema definition*

```
<?xml version = "1.0" encoding = "UTF-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
     targetNamespace = "http://www.ipdr.org/namespaces/ipdr"
     xmlns:ipdr = "http://www.ipdr.org/namespaces/ipdr"
     version = "3.0-A.0"
     elementFormDefault = "qualified"
     attributeFormDefault = "unqualified">
    <include schemaLocation = "IPDRDoc3.0.xsd"/>
    <element name = "clientIP" type="string"/>
    <element name = "startTime" type = "dateTime"/>
    <complexType name = "X_ECHO">
        <complexContent>
            <extension base = "ipdr:IPDRType">
                <sequence>
                    <element ref = "ipdr:clientIP"/>
                    <element ref = "ipdr:startTime"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

LISTING B-2 X_SE_ECHO SCHEMA DEFINITION FILE

- *Screenshots*

FIGURE B-6 X_SE_ECHO MODULE SELECTION

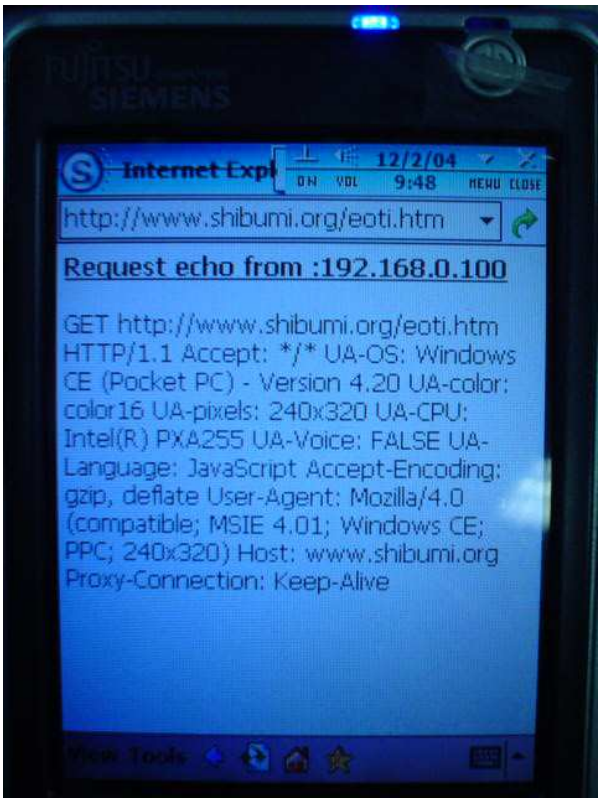

FIGURE B-7 X_SE_ECHO FOR A HTTP REQUEST

x_se_auth

- *Description*: this service allows requests to be authenticated using a specified authentication protocol or method. This service does not run as a stand alone service but is rather used by other service elements that need to enforce client device authentication before handling their requests. An example of a service element that would utilize the `x_se_auth` modules is the `x_se_httpGateKeeper` module.
- *Usage scenario:*
    - A request (`x_req_auth`) for authentication is received.
    - The service element extracts the authentication credentials from the request object.
    - The service element authenticates the provided credentials using the implemented method (e.g. simple database authentication, RADIUS authentication)
    - A response (`x_res_auth`) is constructed and returned to the requesting element
- *Object Diagram*

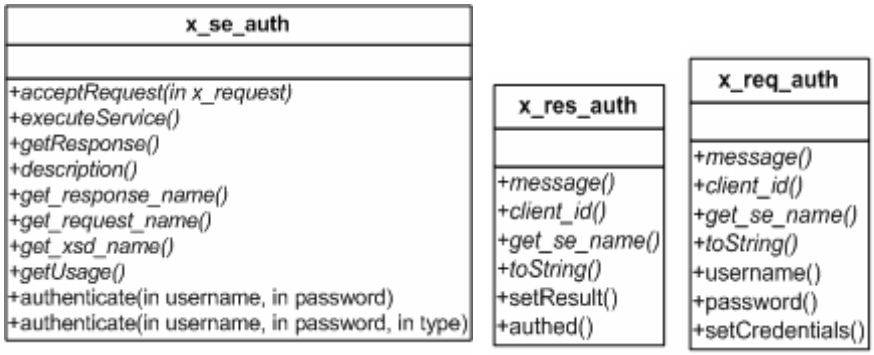

FIGURE B-8 X_SE_AUTH DATA OBJECTS

- *Usage metrics*

Table B-3 `x_se_auth` usage metrics

Category	Usage attribute name	Data type	Presence
Who	ClientIP	String	REQUIRED
When	StartTime	Date/Time	REQUIRED

- *Usage schema definition*

```
<?xml version = "1.0" encoding = "UTF-8"?>
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
     targetNamespace = "http://www.ipdr.org/namespaces/ipdr"
     xmlns:ipdr = "http://www.ipdr.org/namespaces/ipdr"
     version = "3.0-A.0"
     elementFormDefault = "qualified"
     attributeFormDefault = "unqualified">
    <include schemaLocation = "IPDRDoc3.0.xsd"/>
    <element name = "clientIP" type="string"/>
    <element name = "startTime" type = "dateTime"/>
    <complexType name = "X_AUTH">
        <complexContent>
            <extension base = "ipdr:IPDRType">
                <sequence>
                    <element ref = "ipdr:clientIP"/>
                    <element ref = "ipdr:startTime"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</schema>
```

LISTING B-3 X_SE_AUTH SCHEMA DEFINITION

- *Screenshots*

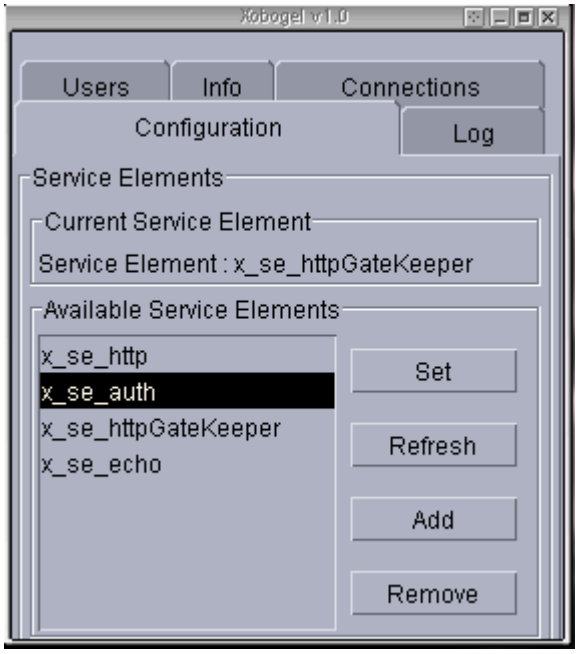

FIGURE B-9 X_SE_AUTH MODULE SELECTION

Appendix C: A Web Browsing Transaction

Web browsing is facilitated by the HTTP protocol which simply allows for formatting and transmission of messages between web servers and browsers. On the OSI reference model, HTTP runs at the presentation layer, on top of TCP at the transport layer and IP at the network layer.

A typical web browsing transaction is depicted in Error! Reference source not found.

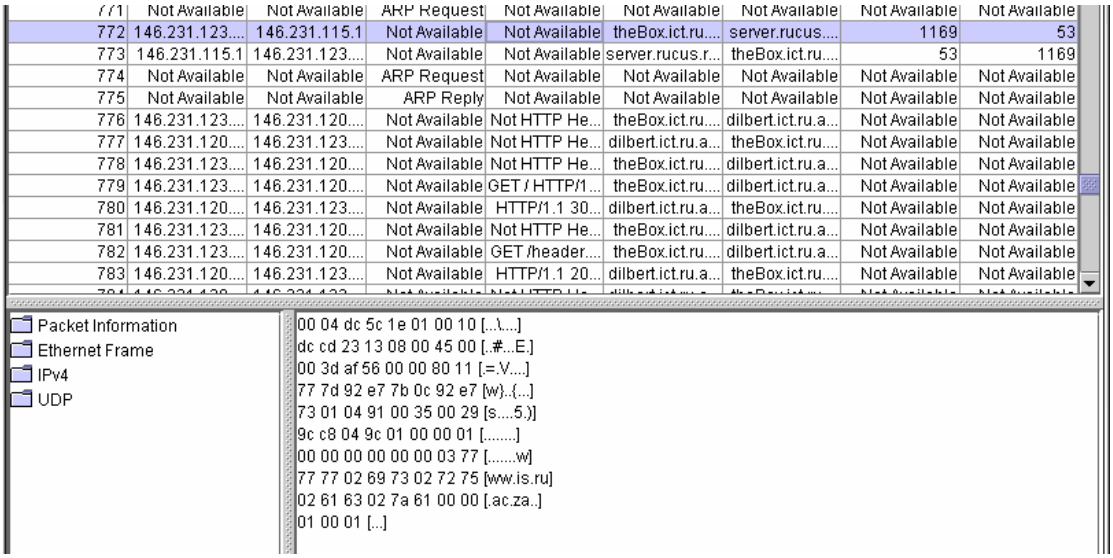

FIGURE C-1 WEB BROWSING TRANSACTION

Sequence of events in web browsing

- A user enters the URL in the web browser **(**Error! Reference source not found.**).**

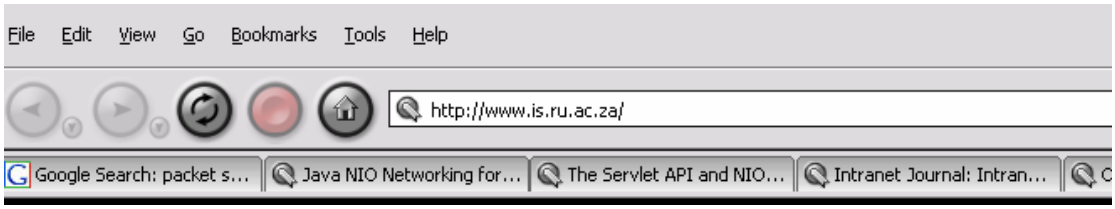

FIGURE C-2 URL REQUEST IN A BROWSER

- In order to send the HTTP request to the right machine, the URL has to be resolved into an IP address and this is done by sending a DNS request to a

DNS server (server.rucus.ru.ac.za) on port 53 **(**Error! Reference source not found.**)**.

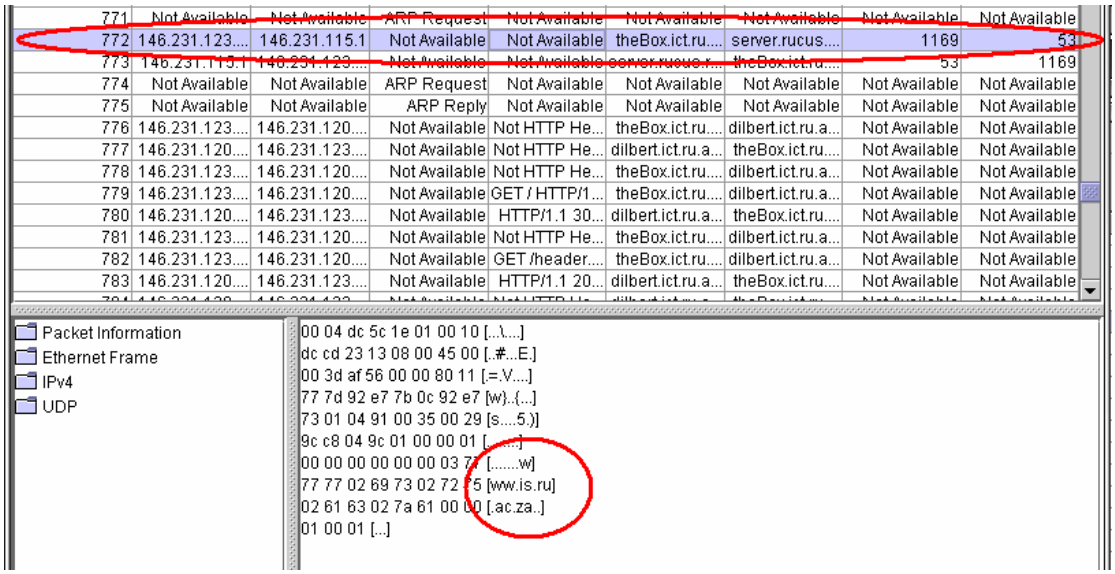

FIGURE C-3 DNS REQUEST

- The DNS server (server.rucus.ru.ac.za) replies with the resolved IP address of the web server host machine **(**Error! Reference source not found.**)**.

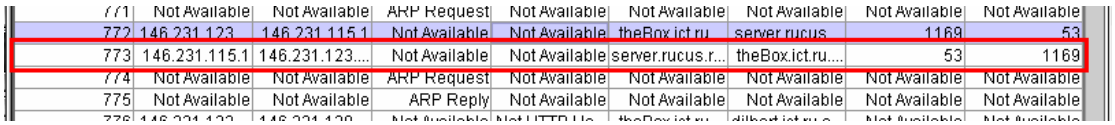

FIGURE C-4 DNS RESPONSE

- The next event is to resolve the IP address into a NIC hardware address (MAC address) using the Address Resolution Protocol (ARP) **(**Error! Reference source not found.**)**. This is done by broadcasting an ARP request. On receipt of the ARP request, the owner of the requested IP address sends an ARP reply that specifies the MAC address.

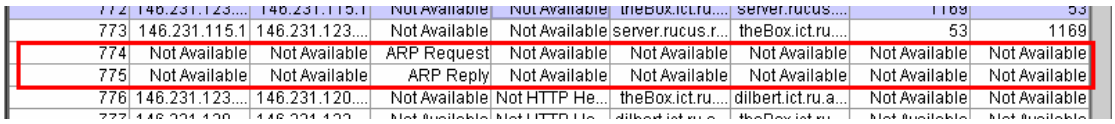

FIGURE C-5 ARP REQUEST/RESPONSE

- This is followed by a three-way handshake between the client (thebox.ict.ru.ac.za) and the web server (dilbert.ict.ru.ac.za) **(**Error! Reference source not found.**)**. In the three-way handshake, the originator sends a SYN packet to establish communication and synchronize sequence numbers, the destination host then sends a SYN/ACK which acknowledges the receipt of the packet and synchronization of the sequence numbers. Lastly the originator sends and ACK packet to acknowledge the receipt of the packet from the destination host.

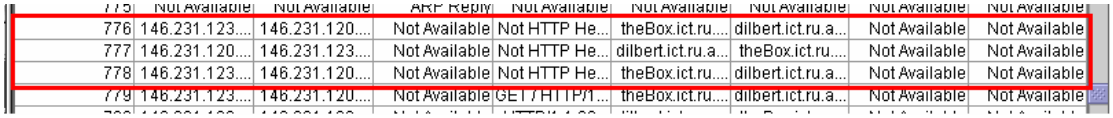

FIGURE C-6 3-WAY HANDSHAKE

- After the three-way handshake, the connection is then open for communication to commence and in this case the first packet send to the destination is a HTTP GET request for www.is.ru.ac.za **(**Error! Reference source not found.**)**.

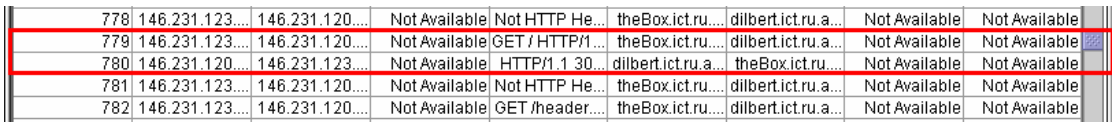

FIGURE C-7 HTTP REQUEST

Appendix D: WISPs questionnaire

EVALUATING THE VIABILITY OF SMALL SCALE, EASILY DEPLOYABLE AND EXTENSIBLE HOTSPOT MANAGEMENT SYSTEMS

[Questionnaire]

---

1. Would you be interested in setting up a hotspot? Yes | No

2. Do you have a computer network currently setup at your premises Yes | No
If yes, have you got a connection to the internet Yes | No

3. What is your occupation (industry)? – Please tick

◯ Information Technology ◯ Education ◯ Food (Restaurants, etc)
◯ Entertainment ◯ Other, please specify _____

4. Please rate your level of computer expertise on a scale of 1 – able to use basic applications, 5 – confident and able to configure a computer and use system applications

1 2 3 4 5

5. What level of security (privacy, confidentiality, data integrity) do you think needs to be implemented on a hotspot (1 – Low, 5 – High)

1 2 3 4 5

6. How important for you is the ability to roam between wireless hotspots. (1 – not important, 5 – very important)

1 2 3 4 5

***Description**: Roaming is the ability to have a vendor-customer relationship with one hotspot provider but to be able to use other hotspots i.e. using single authentication credentials and getting billed on one account. WiFi roaming operates on the same principles as cellular telephony roaming.*

7. What is the approximate average number of clients that you handle per week

◯ Less than 10 ◯ 20 ◯ 30
◯ 50 ◯ 75 ◯ More than 100

8. What operating system do you currently have running on your computers?

◯ Windows ◯ *nix

9. Rate the following internet billing schemes on a scale of most desirable to least desirable (1 – not desirable, 5 – very desirable).

1. Free internet access 1 2 3 4 5
2. Flat rate billing 1 2 3 4 5
1 2 3 4 5

3. Usage based billing

4. Paris Metro Pricing (PMP) (1) (2) (3) (4) (5)

10. Bearing in mind the service provider's interests, e.g. profit making motive, and the user's interests e.g. efficiency, security and quality of service, Rate the following on the basis of which scheme best represents both parties interests, i.e. which is reasonable and the most likely scheme (1 – very unlikely, 5 – very likely).

5. Free internet access (1) (2) (3) (4) (5)

6. Flat rate billing (1) (2) (3) (4) (5)

7. Usage based billing (1) (2) (3) (4) (5)

8. Paris Metro Pricing (PMP) (1) (2) (3) (4) (5)

Description:

- Free internet access is when access is provided as a freebie or an added amenity to another product or service. E.g. free internet access with your cup of coffee, free access at a waiting lounge for SAA passengers.
- Flat rate billing is when a flat charge is levied, say monthly, and then access to the internet is unlimited for that period.
- Usage-based billing is when charging for internet access is based on some usage metric e.g. the amount of time spent on the internet, the amount of data downloaded from the internet.
- Paris Metro Pricing is when the network service is divided into different classes which are charged differently e.g. Class A = R2, Class B = R60, Class C = R80. The users then choose the service class they would like to operate in and they get billed based on that choice. The catch though is that the cheaper classes will invariably be of less quality that more expensive classes mainly due to congestion in those classes.

11. As a service provider, how essential would it be for you to have the flexibility to alter the billing scheme implemented (1 – not essential, 5 – very essential). (1) (2) (3) (4) (5)

12. The two predominant ways of paying for network services are; paying immediately i.e. cash or a coupon, and paying on account i.e. getting billed at the end of the month. As a provider how would you wish to charge for internet access (1 - pay immediately, 3 – do not mind either, 5 - pay on account).

(1) (3) (5)

13. The need for the wireless access Indoors Outdoors
14. Potential interference from closeby access points Yes No
15. The approximate physical coverage

Appendix E: PAL users questionnaire

EVALUATING THE VIABILITY OF SMALL SCALE, EASILY DEPLOYABLE AND EXTENSIBLE HOTSPOT MANAGEMENT SYSTEMS

[Questionnaire]

---

1. Have you ever used a wireless hotspot? (Please circle the right answer) Yes | No

2. Do you own a laptop or a personal digital assistant? Yes | No
If yes, is it enabled to use wireless networks Yes | No

3. What is your occupation (industry)? – Please tick

◯ Information Technology ◯ Education ◯ Health And Medical
◯ Entertainment ◯ Other, please specify _____

4. Please rate your level of computer literacy on a scale of 1 – able to use basic applications, 5 – confident and able to configure a computer and use system applications.

1 2 3 4 5

5. What level of security (privacy, confidentiality, data integrity) do you think needs to be implemented on a hotspot (1 – Low, 5 – High)

1 2 3 4 5

6. How important for you is the ability to roam between wireless hotspots. (1 – not important, 5 – very important)

1 2 3 4 5

***Description**: Roaming is the ability to have a vendor-customer relationship with one hotspot provider but to be able to use other hotspots i.e. using single authentication credentials and getting billed on one account. WiFi roaming operates on the same principles as cellular telephony roaming.*

7. What is the approximate average number of hours that you spend on the internet per week (i.e. web browsing, IRCing, logged on to remote servers, etc)

◯ Less than 1 hour ◯ 5 hours ◯ 10 hours
◯ 20 hours ◯ 30 hours ◯ More than 40 hours

8. How intrusive do you find pop-up adverts during web browsing i.e. do you mind pop-up adverts when you are browsing the internet? (1 – Absolutely do not want pop-up adverts, 5 – do not mind the adverts)

1 2 3 4 5

9. Please tick your top three uses of the internet. (1 – highest usage, 3 – least usage)

◯	E-mailing	◯	Online radio broadcasts
◯	Web Browsing	◯	Instant Messaging
◯	File transfer	◯	Internet telephony
◯	Online gaming	◯	Electronic Commerce
◯	Connecting to company network		
◯	Downloading music		

10. How much would you be willing to pay for higher performance guarantees on a hotspot? (1 – not willing to pay, 5 – willing to pay)

1 2 3 4 5

11. Rate the following internet billing schemes on a scale of most desirable to least desirable (1 – not desirable, 5 – very desirable).

9. Free internet access	1 2 3 4 5
10. Flat rate billing	1 2 3 4 5
11. Usage based billing	1 2 3 4 5
12. Paris Metro Pricing (PMP)	1 2 3 4 5

12. Bearing in mind the service provider's interests, e.g. profit making motive, and the user's interests e.g. efficiency, security and quality of service, Rate the following on the basis of which scheme best represents both parties interests, i.e. which is reasonable and the most likely scheme (1 – very unlikely, 5 – very likely).

13. Free internet access	1 2 3 4 5
14. Flat rate billing	1 2 3 4 5
15. Usage based billing	1 2 3 4 5
16. Paris Metro Pricing (PMP)	1 2 3 4 5

Description:

- Free internet access is when access is provided as a freebie or an added amenity to another product or service. E.g. free internet access with your cup of coffee, free access at a waiting lounge for SAA passengers.
- Flat rate billing is when a flat charge is levied, say monthly, and then access to the internet is unlimited for that period.

- Usage-based billing is when charging for internet access is based on some usage metric e.g. the amount of time spent on the internet, the amount of data downloaded from the internet.
- Paris Metro Pricing is when the network service is divided into different classes which are charged differently e.g. Class A = R2, Class B = R60, Class C = R80. The users then choose the service class they would like to operate in and they get billed based on that choice. The catch though is that the cheaper classes will invariably be of less quality that more expensive classes mainly due to congestion in those classes.

13. As a service provider, how essential would it be for you to have the flexibility to alter the billing scheme implemented (1 – not essential, 5 – very essential).

1 2 3 4 5

14. The two predominant ways of paying for network services are; paying immediately i.e. cash or a coupon, and paying on account i.e. getting billed at the end of the month. As a user how would you wish to pay for internet access (1 - pay immediately, 3 – do not mind either, 5 - pay on account).

1 3 5

References:

(Aboba B. and Zorn G., 1999) – "*RFC 2477: Criteria for evaluating roaming protocols*", January, [Online] Available: http://www.faqs.org/rfcs/rfc2477.html

(Agis E., Mitchel H., Ovadia S., Aissi S., Bakshi S., Iyer P., Kibria M., Rogers C. and Tsai J., 2004) – "*Global, Interoperable Broadband wireless networks: Extending WiMAX technology to Mobility*", Intel Technology Journal, vol 8 - issue 3, August 2004, pp 174

(Aimoto T., 2000) – "*Overview of the Diff-Serv Technology: Its mechanism and Implementation*", IEICE Transcation on Information and Systems (D) E83-D, No. 5, 957-964, May 2000

(Allen E. E., 2001) – "*Diagnosing Java Code: Designing extensible applications*", IBM, [Online] Available: http://www-106.ibm.com/developerworks/java/library/j-diag0925/

(Baer N., 2000) – "*Optimal Web Page Sizes: Is your web site too big?*", 19th October 2000, [Online] Available: http://www.allaboutyourownwebsite.com/bestpagesize.shtml

(Baker K. L., Franz A. M. and Jordan P.W., 1994) – "*Coping with ambiguity in knowledge-based Natural language analysis*", Proc. Of COLING-94, pp 90-94.

(Balachandran A., Voelker M. G. and Bahl P., 2005) – "Wireless hotspots, current challenges and future directions", In: Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, ACM Press (2003) 1–9

(Ballvé F., 1956) – "*Essentials of Economics: A brief Survey of Principles and Policies*", Ch 2, Foundation for Economic Education, Irvington-on-Hudson, New York.

(Blunk L. and Vollbrecht J.,, 1998) – "*RFC 2284: PPP Extensible Authentication Protocol (EAP)*", Network Working Group, [Online] Available: http://www.faqs.org/rfcs/rfc2284.html

(Borisov N., Goldberg I. and Wagner D., 2001) – "*Security of the WEP algorithm*", University of California Berkely, [Online] Available: http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html

(Bridgewater, 2003) – "*Enterprise Wi-Fi Roaming Services: A new revenue opportunity for service providers*", Bridgewater Systems Corporation, December 2003, [Online] Available: http://whitepapers.zdnet.co.uk/0,39025945,60072156p-39000680q,00.htm

(Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M., 1996) – "*Pattern-oriented software architecture: A system of Patterns*". Hoboken, NJ: John Wiley & Sons, 8 August 1996

(Chandra P., 2002) - "*802.11 security*", Wireless Developer Network, [Online] Available: http://www.wirelessdevnet.com/articles/80211security/

(Chao L., 2004) – "*Preface*", Intel Technology Journal, vol 8 - issue 3, August 2004, pp iii

(Choudhari P., 2001) – "*JAVA advantages and disadvantages*", [Online] Available: http://arizonacommunity.com/articles/java_32001.shtml

(Cook J., 2004) – "*Venture Capital: Demise of Cometa sending a message*", Seatle Post-Intelligencer, May 21st, [Online] Available: http://seattlepi.nwsource.com/venture/174310_vc21.html

(Cowan T., 2004) – "*The servlet API and NIO: Together at last*", IBM – Java Technology, [Online] Available: http://www-106.ibm.com/developerworks/library/j-nioserver/

(Crawford W. and Kaplan J., 2003) – "*J2EE Design Patterns*", O'Reilly, September 22, 2003, pp 10

(Cybersource, 2002) – "*Linux vs. Windows – Total cost of ownership comparison*", Cybersource Pty. Ltd, [Online] Available: http://www.e-dynamics.be/docs/linux_vs_windows_tco_comparison.pdf

(Da Silva R., 2000) – "*Billing for GPRS networks*", Telecommunications Development, [Online] Available: http://www.tdap.co.uk/uk/archive/billing/bill(nokia_0009).html

(Deans H., 2004) – "*Born Free: Wild Hotspot Business Model*", Always On, October 6th, [Online] Available: http://www.alwayson-network.com/comments.php?id=4464_0_5_0_C

(Falkner M., Devitsikiotis M. and Lambaradis I., 2000) – "*An overview of pricing concept for broadband IP networks*", IEEE Communications survey, Second quarter 2000.

(Finseth C., 1993) – "*RFC 1492 – An Access Control Protocol, sometimes called TACACS*", Network Working Group, [Online] Available: http://www.faqs.org/rfcs/rfc1492.html

(Fleishman G., 2002) – "*802.11 Task Group Update*", Wireless Devcenter, May 4th, [Online] Available: http://www.oreillynet.com/pub/a/wireless/2002/04/05/80211taskgroups.html

(Floyd S. and Fall K., 1997) – "*Route mechanism to support end-to-end congestion control*", February, [Online] Available: ftp://ftp.ee.lbl.gov/papers/collapse.ps

(Frank R., 2004) – "*Move over, Bluetooth; ZigBee is here*", Design News, March 15, [Online] Available: http://www.designnews.com/article/CA387448.html

(Friedrichs B., 2005) – "*BRAN summary*", [Online] Available: http://portal.etsi.org/bran/Summary.asp

(Geier J., 2002) – "*802.11 Security beyond WEP*", WiFi-Planet, June 26, [Online] Available: http://www.wi-fiplanet.com/tutorials/article.php/1377171

(Geier J., 2003) – "*WPA plugs holes in WEP*", Network World, March 31st, [Online] Available: http://www.networkworld.com/research/2003/0331wpa.html

(GNU, 2005) – "*GNU - Licenses*", Free Software Foundation, June 2005, [Online] Available: http://www.gnu.org/licenses/licenses.html

(Halonen T., Romero J. and Melero J., 2003) – "*GSM, GPRS and EDGE performance*", John Wiley and Sons Inc., November 14, 2003

(Hayes V., 1990) – "*Wireless Access Method and Physical Layer Specifications*", [Online] Available: http://grouper.ieee.org/groups/802/11/Documents/DocumentArchives/1990_docs/11-90017.DOC

(Hernandez C. E., 2000) – "*A pricing scheme for single sources based on the effective bandwidth large buffer asymptotic model*", [Online] Available: http://www.rhsmith.umd.edu/lbpp/jbailey/ents609/EBandHernandez.pdf

(Hunt G. and Thompson R. B., 1998) – "*Windows NT TCP/IP Network Administration : Using Dynamic Host Configuration Protocol*", October 1998, O'Rielly Media, [Online] Available: http://www.intranetjournal.com/articles/200004/im_dhcpb.html

(IBM, 2004) – "*Performance Management Guide : Advantages of Java*", December 2004, [Online] Available: http://publib.boulder.ibm.com/infocenter/pseries/index.jsp?topic=/com.ibm.aix.doc/aixbman/prftungd/java1.htm

(Intel, 2003) – "*Wireless Hotspot Deployment Guide*", Intel in Communications, [Online] Available: http://www.intel.com/business/bss/infrastructure/wireless/deployment/hotspot.pdf

(IPDR, 2005) – "*About IPDR.org*", IPDR.org, [Online] Available: http://www.ipdr.org/about/index.html

(ITFacts-1, 2004) – "*200K hotspots by 2008*", [Online] Available: http://www.itfacts.biz/index.php?id=P1740

(ITFacts-2, 2005) – "*30 mln WiFi users by the end of 2004*", [Online] Available: http://www.itfacts.biz/index.php?id=P747

(JHSPH, 1999) - "*History of wireless*", Available [Online],

(Kokko J., 2003) – "*Mobile Internet Charging: prepaid vs. postpaid*", Network Laboratory, HUT, [Online] Available: http://www.netlab.hut.fi/opetus/s38042/k03/topics/preandpostpaid.pdf

(Kolic R., 2004) – "*Ultra Wideband – the next-generation wireless connection*", DeviceForge, Feb 24, [Online] Available http://www.deviceforge.com/articles/AT8171287040.html

(Lawton G., 2005) – "*What lies ahead for cellular technology*", Computer, vol 38 - issue 6, June 2005, pp 14-17, IEEE computer society press, USA

(Lehr W., 2004) – "*The Economic Case for Dedicated Unlicensed spectrum below 3GHz*", Spectrum Series Working Paper #9, New American Foundation, Washington , DC, July 2004.

(LESSnetworks.com, 2005) – "*Benefits*", [Online] Available: http://www.lessnetworks.com/index.html

(Linderman J. and Bulk F., 2004) – "*Review: wireless LAN security monitors*", Mobilized Software, March 9th, [Online] Available: http://www.mobilizedsoftware.com/showArticle.jhtml?articleId=18311899&pgno=4

(Lloyd B. and Simpson W., 1992) – "*RFC 1334: PPP Authentication Protocols*", Network Working Group, [Online] Available: http://www.faqs.org/rfcs/rfc1334.html

(Mangold S., Choi S., May P., Klein O., Hiertz G. and Stibor L., 2002) – "*IEEE 802.11e Wireless LAN for Quality of Service*", Proc. European Wireless '02, volume 1, pp 32-39, Florence, Italy, February 2002

(Mason R., 2000) – "*Simple Competitive Internet Pricing*", European Economic Review, v44, no.4-6, pp.1045-1056

(McCullagh A. and Caelli W., 2000) – "*Non-Repudiation in the Digital Environment*", First Monday, vol 5 – number 8, August, [Online] Available: http://www.firstmonday.dk/issues/issue5_8/mccullagh/

(Meisner J. and Earnshaw A., 2004) – "*Searching for a profitable business model*", The Business Journal, June 4th, [Online] Available: http://www.bizjournals.com/portland/stories/2004/06/07/story7.html

(Minoli D., 2002) - *"Hotspot Networks – WiFi for public access locations"*, McGraw-Hill Professional Publishing, United States of America, September 2002

(Mitchell B., 2005) – "*Quality of Service*", Wireless Networking, [Online] Available: http://compnetworking.about.com/od/networkdesign/l/bldef_qos.htm

(Mutooni P., 2000) – "*It's time for an IP aware billing model*", Telecommunications Development, [Online] Available: http://www.tdap.co.uk/uk/archive/billing/bill(ibasis_0012).html

(MySQL, 2004) – "*MySQL licensing Policy*", March 2004, [Online] Available: http://www.mysql.com/company/legal/licensing/

(Naeve M., 2005) - "*IEEE 802.15 WPAN ™ Task Group 4*", Jan 19, [Online] Available: http://www.ieee802.org/15/pub/TG4.html

(Needleman R., 2005) – "*Wi-Fi should be free*", CNET Reviews, April 25th, [Online] Available: http://reviews.cnet.com/4520-3000_7-6212563-1.html

(Nocat, 2005) – "*NoCatAuth*", [Online] Available: http://nocat.net/

(O'Hara B. and Petrick A., 1999) – "*The IEEE 802.11 Handbook: A designer's companion*", Standards Information Network IEEE Press, 1999

(Odlyzko A., 2003) – "*Privacy, Economics and Price Discrimination on the Internet*", Fifth International Conference on Electronic Commerce, N Sadeh, ed., ACM Press, pp 355-366

(Patil B., Kulantra S., Saifullah Y., Lyengar L., Faccin S., Sreemanthula S., Aravamudhan L., Sharma S. and Mononen R., 2003) – "*IP in Wireless Networks*", Prentice Hall, January 31, 2003,

(Piana V., 2003) – "*Product Differentiation*", Economics Web Institute, [Online] Available: http://www.economicswebinstitute.org/glossary/product.htm

(PublicIP, 2005) – "*ZoneCD*", [Online] Avaiable: http://www.publicip.net/

(Raven F., 2005) – "*Wireless Last-Mile Technologies Help Bridge the Digital Divide*", EDC Center for Media and Community, April 4, 2005, [Online] Available: http://www.digitaldivide.net/articles/view.php?ArticleID=40

(Reichl P, Stiller B., Ma H., Hasan H., Gerke J., Flury G., 2001) – "*Retrospective Pricing Models for Internet Services: solving the flat rate dilemma 'à la Flensburf'*", Proc. Messung, Modellierung und Bewertung MMB'01, Aachen, Germany, Vol 12, September 2001

(Reynold J., 2003) – "*Going Wi-Fi: A practical guide to planning and building an 802.11 network*", CMP Books, United States of America, September 1, 2003, pp 127

(Rigney C., Willens S., Rubens A. and Simpson W., 2000) – "*RFC 2865: Remote Authentication Dial In User Service (RADIUS)*", Network Working Group, [Online] Available: http://www.faqs.org/rfcs/rfc2865.html

(Scalise K., 1999) – "*Internet Congestion caused by flat-rate pricing and waste, UC Berkeley researchers find*", University of California Berkeley, [Online] Available: http://www.berkeley.edu/news/media/releases/99legacy/5-20-1999.html

(SearchNetworking, 2003) – "*Nomadic Computing*", SearchNetworking.com, April 13, [Online] Available: http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci848042,00.htm

(Sherlekar S. and Prasad R. R., 2003) – "*A global access independent billing model: A user's perspective*", Wireless Personal Communications, Kluwer Academic Publisher, September Issue 2003, pp 169-178

(Shirazi J., 2005) – "*Java Performance Tuning – NIO performance Tips*", [Online] Available: http://www.javaperformancetuning.com/tips/nio.shtml#REF2

(Simpson W., 1994) – "RFC *1994: PPP Challenge Handshake Authentication Protocol (CHAP)*", Network Working Group, [Online] Available: http://www.faqs.org/rfcs/rfc1994.html

(Steiner J.G., Neuman B. C. and Schiller J. I., 1988) – "*Kerberos: An authentication service for open network systems*", Proc. Winter 1988 Usenix winter Conference, pp 191-201, Feb 1988

(Suzuki J. and Yamamoto Y., 1999) – "*Openwebserver: an adaptive web server using software patterns*", IEEE communications, vol. 37, No. 4, pp 46-52, April 1999 http://www.jhsph.edu/wireless/history.html.

(Thinyane M., Foster G. and Clayton P., 2005) – "*Investigating the viability of small scale easily deployable and extensible hotspot management systems"*, Proceedings of South African Telecommunication Networks and Applications Conference 2005, September 2005