UNIVERSIDAD POLITÉCNICA DE MADRID

FACULTAD DE INFORMÁTICA

TRABAJO FIN DE CARRERA

POWER MEASUREMENT AND ANALYSIS OF MOBILE INSTANT COMMUNICATIONS IN 802.11G

MEDICIÓN DE ENERGÍA Y ANÁLISIS DE COMUNICACIONES INSTANTÁNEAS MÓVILES EN 802.11G

AUTOR: José Raúl BENITO SANZ
TUTOR FI-UPM: Prof. Marinela GARCÍA FERNÁNDEZ
TUTOR UNIVERSIDAD EXTERNA: Prof. Antti YLÄ-JÄÄSKI
UNIVERSIDAD EXTERNA: Helsinki University of Technology
PAÍS: Finlandia

ACKNOWLEDGEMENTS

This Master's Thesis has been done for the Telecommunications software and Multimedia Laboratory, in the Department of Computer Science at the Helsinki University of Technology.

First of all, I would like to thank my supervisor, Professor Antti Ylä-Jääski, for giving me the opportunity to work on a research project like this, as well as my instructor, Yu Xiao, for her support and patience, and for all her advices that have been indispensable to finish this thesis.

I also have to be thankful to Francisco and Aldara, my laboratory partners during the development of this work. We worked as a team and really helped each other many times.

Finally, I would like to thank all my friends and my family, for their encouragement and support to have this thesis finished.

Espoo June 2009

José Raúl Benito Sanz

Contents

List of Tables

List of Figures

RESUMEN EN CASTELLANO

Actualmente, el uso de las nuevas tecnologías de información y telecomunicaciones aumenta de forma tan rápida que los teléfonos móviles no solo permiten las típicas llamadas y mensajes de texto, sino que generalmente, están provistos, ya sea mediante redes locales inalámbricas o redes vía satélite, de una conexión a Internet. Uno de los problemas que deben enfrentar los usuarios a la hora de utilizar los servicios que provee Internet es la duración de la batería del aparato que están utilizando, que en el caso del uso para el acceso a Internet se ve notablemente acortada.

Para tener una idea del consumo de energía que requieren los servicios de Internet, se ha elegido un protocolo de mensajería instantánea, dado que este tipo de aplicaciones son de lo más cotidiano, como caso de estudio para determinar cuáles son los factores que más afectan a la duración de la batería.

Hemos dispuesto de un dispositivo Nokia N810, que a pesar de no ser un teléfono convencional ni un *smartphone* (la propia compañía lo llama *internet tablet*), nos sirve para cumplir los objetivos del estudio del consumo de energía.

El propósito principal de este estudio es la determinación de los factores que más afectan al consumo de energía de los dispositivos móviles durante una sesión de mensajería instantánea. Para el análisis de dicho consumo, se procederá a realizar una serie de pruebas bajo una red local inalámbrica. En este entorno, se tendrán controlados varios parámetros, ya sean de la red o de la comunicación instantánea. Dichos parámetros son el tamaño y tipo de los ficheros enviados en una comunicación, intervalos de señalización de la red local inalámbrica (*beacon*), velocidad de transmisión de la red, ubicación de los clientes y la frecuencia y longitud de los mensajes.

Además del análisis de los factores mencionados y de cómo influyen en el consumo energético, el estudio también incluye una breve descripción del protocolo MSNP (*Microsoft Notification Protocol*), utilizado en el popular *Windows Live Messenger*, así como en otros clientes de mensajería instan-

tánea, que ha sido el protocolo utilizado para las pruebas (aunque no con el cliente oficial, sino con uno de código abierto denominado *Pidgin*).

De forma no tan profunda, también se cubren aspectos como el propio dispositivo Nokia N810, el sistema operativo que utiliza, el software utilizado durante las pruebas, la red inalámbrica usada, etc.

Durante el desarrollo del estudio se explicarán las cuestiones básicas relativas al protocolo MSNP, con el fin de conocer qué tipo de paquetes se intercambian clientes y servidor para establecer y mantener una comunicación instantánea. Posteriormente, se describirán con detalle los experimentos a realizar durante las pruebas, así como qué factores se mantendrán fijos y cuáles serán variables durante la realización de las mismas, de forma estructurada en una serie de casos de uso. Tras ello, se presentarán los resultados de las pruebas, describiendo los aspectos más interesantes, así como definiendo en función de cuáles de los factores analizados varía el consumo energético.

Además de la descripción de las pruebas y sus resultados, también se realiza un recorrido por el trabajo previo en lo relativo a mensajería instantánea y consumo energético (o más correctamente, ahorro de energía).

En lo relativo a la mensajería instantánea, los estudios prestan atención al desarrollo de protocolos que permitan este tipo de comunicación y de las aplicaciones que los utilicen. De estos dos aspectos, el que requiere una mayor visión es el diseño del protocolo. Dada la gran cantidad de protocolos que hay disponibles, hay una cierta controversia a la hora de decidir cuál es mejor y cual peor, debido a las ventajas y desventajas de unos y otros. Además de ello, ciertos estudios han intentado establecer un protocolo estándar, con resultado negativo. Fuera de aspectos puramente técnicos, hay estudios sobre cómo influye la mensajería instantánea en los negocios o incluso en las relaciones personales.

Respecto al ahorro de energía, la investigación actual se centra principalmente en el software, aunque las compañías distribuidoras de teléfonos y otros dispositivos móviles también tratan de mejorar el hardware (las baterías principalmente).

En la última parte del estudio, se exponen las conclusiones obtenidas a partir de los casos de prueba. El factor principal a la hora de influir en el consumo energético es el tiempo, como se podrá ver. Sin embargo el tiempo invertido en las comunicaciones es afectado por prácticamente todos los demás aspectos que se regulan durante las pruebas. Además del tiempo, se podrá comprobar que en el momento en el que cualquier tipo de evento sucede en el dispositivo móvil (tocar la pantalla táctil, presionar una tecla) tiene repercusión en el consumo de energía, aunque siempre acotado bajo un máximo, al menos en los casos estudiados, en los que ningún otro proceso aparte del cliente de mensajería instantánea estaba activo. Volviendo a la cuestión del tiempo invertido en las comunicaciones, podremos ver cómo el tamaño de los ficheros enviados a través de una comunicación es un factor decisivo. Los ficheros más grandes requieren un tiempo mayor para que se complete el envío, derivando ello en un mayor consumo total de energía. En los mismos casos de prueba en los que se analiza el tamaño de los ficheros, también se analizará el tipo de los mismos, aunque los resultados mostrarán que este factor no es determinante en ninguno de los casos, puesto que cualquier fichero es encapsulado en paquetes que no tienen en cuenta la estructura interna del mismo.

Para los casos de prueba que tienen en cuenta el intervalo de las señales de la red inalámbrica. Un intervalo de señalización muy corto forzará a los dispositivos a capturar y analizar esas señales de forma continua. Aunque el procesamiento de estas señales de *beacon* es completamente transparente al usuario del dispositivo, requiere una cierta actividad por parte de la CPU que, obviamente, consume energía. Tal consumo será mayor cuanto más corto sea el intervalo. Poniéndonos en el caso completamente opuesto, en el que los intervalos de señalización son muy amplios, no se obtiene un mejor rendimiento en cuanto a ahorro de energía. Para este caso, se explicará el concepto de *modo de ahorro de energía*, mediante el cual, el dispositivo móvil permanece en un estado de bajo consumo mientras no tiene que procesar los paquetes de señalización de la red. Al incrementarse el tiempo entre estos paquetes, el dispositivo pasa mucho más tiempo ahorrando energía que interaccionando con la red, lo cual aumenta notablemente los tiempos de transferencia de datos y con ello, la energía consumida en total.

Otro de los factores que se analizarán será la velocidad de transmisión de la red. Como se podrá ver, aunque existen ciertas diferencias entre el consumo observado con unas y otras velocidades, este factor no resulta tan determinante a la hora de realizar el consumo de energía total.

También se podrá comprobar que la ubicación de los clientes de mensajería (pudiendo estar en la misma red o en redes distintas) tiene un efecto menor en el consumo energético. La comunicación es mucho más rápida si ambos clientes están conectados a la misma red desde el mismo punto de acceso, lo que provoca menores retardos, menor tiempo total de las comunicaciones y, en definitiva, menor consumo energético, pero las otras situaciones analizadas, en las que los dispositivos se encuentran en redes distintas, aunque ven aumentado su consumo, no se dispara como pudiera ocurrir con tamaños de información demasiado grandes. Además, hay que notar que, por norma general, en una comunicación instantánea, emisor y receptor de los mensajes están en redes separadas, por lo que no es un parámetro que se pueda controlar.

Finalmente, el análisis de la frecuencia y la longitud de los mensajes enviados y recibidos, además de ser el más representativo, puesto que las aplicaciones de mensajería instantánea se usan principalmente para el intercambio de mensajes y no de ficheros, es el que muestra unos resultados más reveladores. Por otra parte, también muestra grandes diferencias entre los casos en que los mensajes son recibidos y entre los que envían mensajes. Cuando es el dispositivo móvil el que recibe los mensajes, la longitud del mensaje es prácticamente irrelevante a la hora de medir la energía consumida. Esto sucede debido a la limitación del mensaje que ofrecen las aplicaciones y que, en la mayoría de ellas, no se permite que un mensaje tenga más longitud que la del paquete donde va a ser encapsulado, por lo que la diferencia entre el consumo que provoquen unos mensajes cortos y otros largos está determinada por unos pocos bytes. El comportamiento es muy distinto a la hora de enviar un mensaje desde el dispositivo móvil, ya que por cada carácter que contenga el mensaje, habrá sido necesaria la pulsación de una tecla o un contacto en la pantalla táctil, eventos que sí tienen una gran repercusión en el consumo de energía. En este caso, la longitud del mensaje sí es determinante a la hora de ahorrar energía. Por otro lado, también influye la frecuencia con la que se envían los mensajes. Un envío continuo de mensajes provoca un mayor consumo que el enviar mensajes de forma espaciada. Respetando unos ciertos intervalos de tiempo entre mensaje y mensaje, el dispositivo puede de algún modo descansar (dependiendo de la configuración del usuario, incluso oscurece o apaga la pantalla, reduciendo el uso de la batería) y ahorrar energía. Combinando ambos aspectos, longitud y frecuencia de los mensajes, se observará que el mejor rendimiento en cuanto a ahorro de energía se consigue con mensajes largos y poco frecuentes.

Finalmente, se darán unas pequeñas nociones sobre lo que podrían ser futuras investigaciones en el área tratada, como por ejemplo, el diseño de software que trate de evitar las peores condiciones para el consumo energético, como el envío de ficheros extremadamente grandes, elevados intervalos de señalización, etc.

ABBREVIATIONS AND ACRONYMS

1G	First Generation Wireless Telephone Technology
2G	Second Generation Wireless Telephone Technology
APT	Advanced Packaging Tool
CRC	Cyclic Redundancy Checking
CSV	Comma-Separated Values
DDR	Double Data Rate
EDR	Enhanced Data Rate
GPRS	General Packet Radio Service
GPS	Global Positioning System
GUI	Graphical User Interface
HW	Hardware
IEEE	Institute of Electrical and Electronics Engineers
IMAP	Internet Message Access Protocol
IP	Internet Protocol
LAN	Local Area Network
MSNP	Microsoft Notification Protocol
NIC	Network Interface Card
OFDM	Orthogonal Frequency-Division Multiplexing
PDA	Personal Digital Assistant
PND	Personal Navigation Device
POP	Post Office Protocol
RAM	Random Access Memory
RSS	Really Simple Syndication
SD	Secure Digital (memory card)
SDHC	Secure Digital High Capacity
SMTP	Simple Mail Transfer Protocol

SSID	Service Set Identifier
SW	Software
TI OMAP	Texas Instruments Open Multimedia Application Platform
TIM	Traffic Indication Map
UMPC	Ultra-Mobile Personal Computer
UPnP	Universal Plug and Play
URL	Uniform Resource Locator
USB	Universal Serial Bus
VGA	Video Graphics Array
WLAN	Wireless Local Area Network
WVGA	Wide Video Graphics Array

Chapter 1

INTRODUCTION

The use of wireless communication technologies has been increasing since several years ago, and it is continuously spreading more and more. The first commercial launch of a mobile phone occurred in Japan in 1979. It was the beginning of the first generation of mobile phones (1G), although the massive use of this devices did not happen until the mid 1990s, with the second generation (2G). Nowadays, the use of cellphones is so extended that even young people or kids have their own phone. In addition, the objective of cellphones nowadays not only is to provide voice calls or text messages to the users, but also multimedia applications (such as images, music and videos), video games, Internet connection and so on. Therefore, this devices are called "smartphones", unlike just "mobile phones" or "cellphones".

Apart from these smartphones, there are some other devices similar to them, but without the phone features like voice calls or text messages. They are the "Internet tablets", devices quite close to PDAs but usually with additional features. In this thesis, all the work has been carried out with a Nokia N810 tablet.

One of the problems of these devices is the battery lifetime, notably shorter in the smartphones and Internet tablets than in older mobile phones. According to the specifications given by Nokia about the battery of the N810 tablet, the lifetime is the following:

- **Continuous usage.** I.e. display light on and wireless LAN active, up to 4 hours.
- **Music playback.** Up to 10 hours.
- **Always online time.** Up to 5 days.

- **Standby time.** Up to 14 days.

This means that the standby and online time are ok, but if the user tends to use the device continuously, the battery lifetime is not long enough and it will be mandatory to recharge the battery every day.

This thesis' aim is to determine the factors that have most influence on the power consumption.

1.1 Problem Statement

As said before, the purpose of the thesis is to determine the factors that affect most the power consumption of the tablet while running an instant messenger program.

To analyze the power consumption, there will be a series of tests consisting on sending several files from one user to another, and keeping chat sessions while the power is measured.

The factors that will be analyzed will be the following:

- Size of the files sended or received.
- Type of the files sended or received.
- Beacon interval of the WLAN.
- Transfer rate of the WLAN.
- Peer locations of the message sender or receiver.
- Frequency and length of text messages sended or received.

1.2 Scope

The main contents of the thesis is the power consumption analysis on the N810 tablet, depending on the factors mentioned above. This analysis includes the study of the protocol used in MSN Messenger, the Microsoft Notification Protocol (MSNP).

Other aspects also covered are the wireless network, the N810 device and different software used, but not as deeply as those mentioned above. The analysis of the wireless network only concerns the issues that most affects the power consumption, without paying attention on the details of the network architecture or intern functioning, as well as with the N810 and the software used. Just brief descriptions about them are included, regardless the design of both. Furthermore, the software used is already implemented and available. No new software has been developed for the purpose of this thesis.

1.3 Structure of the Thesis

This thesis is divided into several sections. First of all, the basics of the **MSNP protocol** will be explained, in order to know what kind of packages are sent to establish an maintain an instant communication with a messenger application.

Next chapter will focus on the **experiments** used in the study, with a brief reference to the N810 device used and the necessary software to make the measurements, although the main point of this part is to describe the use cases in order to determine which factors are to be analyzed and why.

Later on, we will explain the **results of the experiments**, describing the aspects each one covers, and **analyzing** these aspects to build the power consumption models.

Finally, the **conclusions** and future work are explained.

Chapter 2

RELATED WORK

This chapter will try to review the previous work published by other researchers which is related to the topics of this study, stating the differences and similarities. Therefore, the main aspects will be mobile devices, instant messaging and power saving methods.

2.1 Introduction

To begin with, we will take a look on different studies based on mobile devices. Either phones, PDAs, GPS systems or any other kind of machine, mobile devices have become more complex throughout the years, requiring better hardware and especially software. The most clear example is with mobile phones, which several years ago only allowed to make phone calls with a simple interface, and now are complete terminals, almost like laptop computers with their own operating system and full interaction with any other device in a computer network.

Therefore, research in mobile devices have covered (and still it will) many aspects, as operating systems, GUI design, security, network interaction, Internet browsing, etc. as well as many applications that could be run in any all-purpose computer. However, although a mobile device has been the main tool in the development of this study, none of the aspects mentioned above has been the purpose of the project.

2.2 Related Work on Instant Messaging

Now focusing on a less general point of view than the "mobile device" concept, research on instant messaging has had a major role in the recent years. Allowing people to communicate in an easy and cheap way, we have witnessed how different instant messaging technologies raised, such as AIM, ICQ, Jabber, Skype, Yahoo! Messenger, and (the one related to this study) MSN Messenger, now known as Windows Live Messenger.

The development of a IM technology is not an easy process, despite there are lots of different protocols and applications as those mentioned before. Using IM technology involves two main points: An application to allow the users to communicate which requires a whole software engineering process, and the design of a protocol to manage the communication. Among these two, the one that requires a more exhaustive research process is the design of the protocol. Especially about this last one, we can find interesting studies analyzing and comparing different protocols and systems even making distinctions between mobile and fixed networks. Regarding the different IM protocols, there are also studies trying to establish a standard protocol. Advantages and disadvantages of IM are covered in different studies too, depending on certain characteristics that make one protocol better or worse than another one [21], or just determining if the use of IM system is worth compared with other communication methods [3]. In this thesis, there is no comparison between different protocols or communication methods. The study just covers a part of the MSN protocol in order to determine the functioning, but the main objective is not the study of the protocol, but the power consumption related to it. A good topic to continue the work of this thesis could be the comparison between protocols and decide which one among them is more energy-efficient.

About the different instant messaging software clients, we can find studies focusing on building a complete software application, which will use a given protocol, so the work is entirely focused on the software design and implementation [25]. A step beyond related to developing IM software applications is to take advantage of any other characteristic offered by the hardware (or even software) in which the IM client will be run. For example, using a GPS equipped hardware (e.g. a mobile phone with this technology), it is possible to build a position-aware IM client [5], what introduces the concept of cross-platform applications. Anyway, in this thesis there is no software development. A software already available and tested (Pidgin) has been used, trusting its reliability.

Research on instant messaging can cover lots of aspects, even outside the fields of computer science, like studies related to IM in business, or how it affects interpersonal relationships, among so many other studies focusing on the social aspects [27]. Another field of study concerning IM, but not related to software technology and similar is the language, as we can find studies analyzing the manners in which users communicate with each other, comparing spoken and written language [14]. But anyway, these studies are not closely related to the purpose of this study, nor even related to computer science engineering.

2.3 Related Work on Power Saving

Finally, the topic in which this study most focuses, the power saving, has become very important in the recent years, especially in mobile devices. It is obvious that in devices like desktop computers and such, the power consumption is not a major problem, since the user always has a power supply which will not prevent him or her to stop the activity if the power runs out. However, when using a mobile device, the power supply becames crucial, since the user may have to stop his or her activity if the device runs out of energy. Therefore, research in this field becomes extremely important, especially for the manufacturers and vendors of this kind of devices, since big differences in this aspect may cause higher or lower economic benefits in the end.

In order to achieve a lower power consumption on this devices, research may focus either in the hardware, e.g., different battery systems depending on the materials, circuitry, etc., or in the software, preventing the applications of the devices to make an excessive use of the hardware if the conditions are not suitable.

A possible way to improve the power saving of any electronic device is the use of a better battery, i.e., a battery which consumes less electrical energy. For example the design of an entire power-saving transformer [2]. This is a good method to improve the power saving, since the design of this electrical batteries or transformers can be applied to any device like PDAs, mobile phones and such, but anyway, this kind of research goes to a very low level compared to this study, which does not pay attention to the internal circuitry of the device.

The present study is more related to the software research more than the hardware. What the thesis presents in this aspect is a deep analysis on the characteristics that cause a higher power consumption in a mobile device, although it does not develop any power saving method. There are several studies covering power saving aspects, either in the application protocols (like MSNP in our case) or in lower layers of the protocol stack. In this case, the improvements would aim to transport or network layer.

An example of protocol design to achieve a better energy consumption is the following. Based on a wireless sensor network, data is sent through different stations, either static or mobile, in order to measure the energy consumption differences [22]. This protocol's study is similar to the this, in a way that it determines which settings are more suitable for a lower consumption, but it focuses on lower layers of the protocol stack. Considering that the applications used in any environment can cause different levels of activity, and therefore different consumptions, this thesis focuses on a single application layer protocol, widely used, and determines its behaviour and possible ways to improve the energy saving methods.

Another topic related to this thesis is the power modeling. As will be seen in the results of the experiments, the power consumption is, in most cases, directly proportional to certain parameters. Research on power modeling is also a widely covered topic, as we can see in [23]. That study covers both power modeling and fault tolerance topics, although not related to a computer-based device. In this thesis the fault tolerance is not covered, but the power modeling is. The main difference is that this study bases its power modeling on software characteristics, and the other one is based on physical characteristics, since the research is based on e-textiles.

Beside the previous work mentioned. The topic in which most work can be found is the development of energy-aware applications. The idea is to design and implement a software dedicated to a certain work, but comparing it to any other software that is already developed, and making the new one more energy efficient. Research on this area can cover memory and processing elements in embedded systems [15], dynamic software management independent from the operating system [8], framework for mobile computing [4], services for sensor networks [16], and such. The main difference with this thesis is that here, we focus on a concrete application and protocol, and we study the characteristics that affect the power consumption of this kind of protocol and application (and also some others that deal with the network parameters), instead of developing a more general system to improve the energy consumption.

Chapter 3

INSTANT MESSAGING

In this chapter, we will explain quite deeply the instant messaging protocol used in the study, the MSNP, which stands for Microsoft Notification Protocol, most known as MSN protocol. Although Pidgin supports other IM protocols, MSNP was chosen since it is the most used worldwide.

The chapter will try to cover as many aspects of the MSNP as possible, but it will focus on those more relevant to the study, such as file transfers and instant messages.

3.1 MSNP Overview

First of all, to clarify some concepts. The term “MSN Messenger” is commonly used to refer to the program and the network used in the instant communications, but in fact, “MSN Messenger” is just the application used to chat inside the “MSN Messenger network”, and this program uses the “MSN Protocol”. This “MSN Messenger network” was launched in 1999 and it belongs to Microsoft.

The way the protocol works consists on a client-server communication. The program installed in a computer is always a client, and it communicates with other clients via a server. In fact, each client is only connected to the server, and this server is in charge of receiving messages from one client and send them to another, so the communication is established properly.

Here, the term“messages” not only refer to written text by an user who wants to chat with another one, but also other information required to control the communication, like contact details, status information and such.

Since the program used in this study was not the “official” MSN Messenger, it was Pidgin, we will simply refer to the program as “the client”, except when we want to compare it with MSN Messenger, which will be referred as “the official client”.

Basically, the MSN protocol consists on a series of commands sent between the client and the server. Most of the commands are sent from the client to the server, and then the server may give a response to the first client, or forward some other command to a different client. There are also some commands that are sent firstly from the server to the client, and then the client replies to the server.

The protocol works as both instant messaging and presence system. “Presence” is the status the person has while connected (or not) to the system, like “online”, “busy”, etc. and of course “disconnected”. An IM session involves a connection to a notification server, which provides presence service, and to a switchboard server, which provides instant messaging service.

3.2 MSNP Commands

All the information sent between server and clients is in the form of a command. The basic structure of a normal command is the following:

- A three capital letters **code**.
- A **transaction ID**, explained later.
- The **parameters** of the command, which vary depending on the code.
- A **new line**, also explained later.

These normal commands are mainly sent by the client and they cause the server to respond with one or more commands.

Also there are some different types of commands, consisting on:

- **Payload commands.** These commands span multiple lines and they do not have universal format. It depends on each specific command. Message commands (MSG) are included in this cathegory.
- **Error commands.** These are sent by the server when it is not able to respond to a client command (either because it could not obey the

order or understand it). Instead of a command code, error commands always have a three-digit code with no parameters.

- **Asynchronous commands.** These commands are sent by the server, but without being a response to the client. For example, when a contact in our list gets connected to the network, the server notifies it with an asynchronous command.

3.2.1 Newlines, Parameters and Transaction IDs

Every normal command must end with a newline. Most of the clients end the commands with **\r\n**, although the servers may accept commands ended only with **\n**. Newlines are necessary to notify the server when a command ends.

Parameters provide additional (but necessary) information to the commands. Most of the commands have one or several parameters, e.g. when a contact gets connected or changes his or her status, the server will send a **NLN** command with, among other parameters, the address of the contact. Every parameter is separated from the previous one (or from the command code) by a blank space.

Transaction IDs (TrIDs from now on) are a special kind of parameter. Their purpose is to match a client command with a server response.

Let's take a look on an example:

52	16.351568	192.168.1.104	messenger.hotmail.msn	MSNMS	VER 1 MSNP9 MSNP8 CVR0	49527	msnp
53	16.530231	messenger.hotmail.msn	192.168.1.104	MSNMS	VER 1 MSNP9	msnp	49527
55	16.532738	192.168.1.104	messenger.hotmail.msn	MSNMS	CVR 2 0x0409 winnt 5.1 i386 MSNMSGR 6.0.0602 MSMSGS tmltest01@h	49527	msnp
56	16.711472	messenger.hotmail.msn	192.168.1.104	MSNMS	CVR 2 8.1.0178 8.1.0178 8.1.0178 http://msgruser.dlservice.micr	msnp	49527
57	16.713300	192.168.1.104	messenger.hotmail.msn	MSNMS	USR 3 TWN I tmltest01@hotmail.com	49527	msnp
58	16.891994	messenger.hotmail.msn	192.168.1.104	MSNMS	XFR 3 NS 207.46.109.101:1863 0 207.46.28.94:1863	msnp	49527

Figure 3.1: Example of commands and responses with their respective TrIDs.

As seen in the figure, the TrID is the first parameter appearing right after the command code. It is always a positive number, starting from 1 (although 0 is also a valid TrID, it is only used with certain asynchronous commands).

In the example shown above, the command **VER**, establishes the protocol version that will be used in the communication. Once the server receives the client message, it replies with another **VER** command with the same TrID. The **CVR** command below delivers information about the client application and operating system, and it is also followed by another **CVR** with the same TrID. Finally, the **USR** command, which is used to notify the server about the address of the account which is logging in, receives a different command,

XFR, but with the same TrID. Later on we will explain the purpose of each command.

3.3 Notification, presence and pings

This section covers mostly the aspects related with the "notification stage" of the MSN protocol. These aspects are those like authentication, getting contact lists or contacts logging in. I.e., they are situations where there is a communication between the client and the server, but the do not involve messages sent between the client users. Also the presence information (commonly the contacts' status) and the keep-alive signals (pings) between the client and the server are commented here.

Below is an example of a typical session using an MSN client.

52	16.351568	192.168.1.104	messenger.hotmail.msn	MSNMS	VER 1 MSNP9 MSNP8 CVR0	49527	msnp
53	16.530231	messenger.hotmail.msn	192.168.1.104	MSNMS	VER 1 MSNP9	msnp	49527
55	16.532738	192.168.1.104	messenger.hotmail.msn	MSNMS	CVR 2 0x0409 winnt 5.1 i386 MSNMSGR 6.0.0602 MSMSGS tmltest01@h	49527	msnp
56	16.711472	messenger.hotmail.msn	192.168.1.104	MSNMS	CVR 2 8.1.0178 8.1.0178 8.1.0178 http://msgruser.dlservice.micr	msnp	49527
57	16.713300	192.168.1.104	messenger.hotmail.msn	MSNMS	USR 3 TWN I tmltest01@hotmail.com	49527	msnp
58	16.891994	messenger.hotmail.msn	192.168.1.104	MSNMS	XFR 3 NS 207.46.109.101:1863 0 207.46.28.94:1863	msnp	49527
66	17.152527	192.168.1.104	207.46.109.101	MSNMS	VER 4 MSNP9 MSNP8 CVR0	57146	msnp
67	17.348082	207.46.109.101	192.168.1.104	MSNMS	VER 4 MSNP9	msnp	57146
69	17.351069	192.168.1.104	207.46.109.101	MSNMS	CVR 5 0x0409 winnt 5.1 i386 MSNMSGR 6.0.0602 MSMSGS tmltest01@h	57146	msnp
70	17.546686	207.46.109.101	192.168.1.104	MSNMS	CVR 5 8.1.0178 8.1.0178 8.1.0178 http://msgruser.dlservice.micr	msnp	57146
71	17.549259	192.168.1.104	207.46.109.101	MSNMS	USR 6 TWN I tmltest01@hotmail.com	57146	msnp
72	17.745308	207.46.109.101	192.168.1.104	MSNMS	USR 6 TWN S ct=1219150038,rver=5.5.4177.0,wp=FS_40SEC_0_COMPACT	msnp	57146
123	20.567327	192.168.1.104	207.46.109.101	MSNMS	USR 7 TWN S t=97hCEnbXmEQme1*mxN2DIQRCqBqUI!LOc*48*E6afv73!899B	57146	msnp
126	20.801887	207.46.109.101	192.168.1.104	MSNMS	USR 7 OK tmltest01@hotmail.com Researcher 1 0	msnp	57146
127	20.802328	207.46.109.101	192.168.1.104	MSNMS	MSG Hotmail Hotmail 519	msnp	57146
130	20.819456	192.168.1.104	207.46.109.101	MSNMS	SYN 8 0	57146	msnp
131	21.107263	207.46.109.101	192.168.1.104	MSNMS	SYN 8 1 1 2	msnp	57146
132	21.110890	207.46.109.101	192.168.1.104	MSNMS	GTC A	msnp	57146
133	21.113934	207.46.109.101	192.168.1.104	MSNMS	PRP MBE N	msnp	57146
134	21.117256	207.46.109.101	192.168.1.104	MSNMS	LSG 0 Individuals 0	msnp	57146
135	21.120303	207.46.109.101	192.168.1.104	MSNMS	LSG 1 Buddies 0	msnp	57146
136	21.125026	207.46.109.101	192.168.1.104	MSNMS	LST tmltest00@hotmail.com Researcher 11 1	msnp	57146
138	21.147402	192.168.1.104	207.46.109.101	MSNMS	CHG 9 NLN 1073741856	57146	msnp
139	21.343639	207.46.109.101	192.168.1.104	MSNMS	CHG 9 NLN 1073741856	msnp	57146
140	21.345256	192.168.1.104	207.46.109.101	MSNMS	BLP 10 BL	57146	msnp
142	21.588191	207.46.109.101	192.168.1.104	MSNMS	BLP 10 1 BL	msnp	57146
144	40.666978	207.46.109.101	192.168.1.104	MSNMS	CHL 0 32454309972279983472	msnp	57146
146	40.679165	192.168.1.104	207.46.109.101	MSNMS	QRY 12 PROD0038W!61ZTF9 32	57146	msnp
147	40.874960	207.46.109.101	192.168.1.104	MSNMS	QRY 12	msnp	57146
149	43.637214	207.46.109.101	192.168.1.104	MSNMS	NOT 597	msnp	57146
151	51.355199	192.168.1.104	207.46.109.101	MSNMS	PNG	57146	msnp
152	51.625553	207.46.109.101	192.168.1.104	MSNMS	QNG 43	msnp	57146
154	81.357395	192.168.1.104	207.46.109.101	MSNMS	PNG	57146	msnp
155	81.630504	207.46.109.101	192.168.1.104	MSNMS	QNG 40	msnp	57146
178	211.848742	207.46.109.101	192.168.1.104	MSNMS	NLN NLN tmltest00@hotmail.com Researcher 44 0	msnp	57146
203	277.741366	207.46.109.101	192.168.1.104	MSNMS	FLN tmltest00@hotmail.com	msnp	57146
210	312.625757	192.168.1.104	207.46.109.101	MSNMS	OUT	57146	msnp

Figure 3.2: MSNP packets sniffed in a typical session.

The figure shows only the MSNP packets sniffed with Wireshark during a test session. Packets from other protocols are not shown.

What the user does (or is notified) in the session above is very little. Just connecting his or her client to a server (logging in), waiting for a contact to connect, waiting for this contact to disconnect and then closing the connection. With this four steps we can take a deep look on what communications the protocol establishes.

A deep description of all the commands observed in the experiments is shown in the appendix at the end of the document.

3.4 Communication

This section deals with the aspects related to the communication in the MSN protocol, i.e., the instant messages and the file transfers.

In order to communicate one client with another, before starting sending messages, a switchboard session must be created. A switchboard session is nothing more than a chat, but internally, the MSN protocol must perform certain commands exchanges before allowing the text messages between users.

Older versions of MSNP used to have a file transfer protocol called MSNFTP. However, with the version of the protocol that Pidgin uses, the behaviour is completely different. In fact, a whole file is sent as if it was an enormous text message, split in many different parts, each one included in a single message packet.

A full explanation of the switchboard sessions, both for message exchanges and file transfers is provided in the appendix at the end of the document.

Chapter 4

EXPERIMENTAL SETUP

This chapter explains the most important aspects about the wireless networks concerning the purpose of the study, i.e., the power consumption of a device using an IM software. Also a brief description of the device used and its operating system are shown.

4.1 Nokia N810 and Maemo Operating System

The concept "internet tablet" is used exclusively by Nokia to refer to certain mobile devices that provide access to the Internet, among so many other applications, similar to those available with PDAs. The device used in the study is the Nokia N810. The Nokia N810 device is an "internet tablet" from Nokia, announced on 17 October 2007. This device does not allow to make phone calls, but to connect to the Internet using Wi-Fi networks or exchange information with a mobile phone via Bluetooth. In this study, the bluetooth connection has not been used, so we will focus on the Wi-Fi connection. Concerning the user interface, the Nokia N810 is composed of a touchscreen (to be tapped with a stylus) and a keyboard. It includes GPS, what makes the N810 not only a PDA and a media player, but also a PND (Personal Navigation Device) and a small UMPC. N810 features the Maemo Linux distribution (explained later in this chapter), including MicroB (a Mozilla-based web browser for Maemo), a GPS navigation application and a user-friendly interface, based on the touchscreen. Since Maemo is one more Linux distribution, it allows the installation of many other applications with a simple "apt-get" command in the Linux-like shell, e.g., Pidgin, the instant messaging program that was used in this study.

Formerly named “Internet Tablet OS”, Maemo is the operating system running in the Nokia N810. The version used in this project is 4.1.2, alternatively named OS2008, stable since May 2008. Maemo can be considered another Linux distribution, but oriented to a handheld device. It is based on Debian GNU/Linux, using much of its GUI, frameworks and libraries from the GNOME project. Like other Debian-based operating systems, it can be easily updated and software can be installed using the APT application. Although most of the applications of Maemo are open source, a few components may be closed source, particularly the Wi-Fi drivers. Some other differences with other Linux distributions are *BusyBox*, a software package that replaces the GNU Core Utilities, requiring less memory and storage in order to adapt the workload to a device with less capabilities, and only a single window at a time in the screen (although the operating system is preemptive multitasking), among others.

More information about the N810 device and Maemo Operating System can be found in the Nokia Website.

4.2 WLAN Outstanding Aspects

As written in the N810 description, the ways this device has to communicate with others are either a Bluetooth connection or a WLAN connection. Since the one used in this thesis is the WLAN, now we will try to explain some of the most important aspects of the WLAN connections, especially those who deal with the power consumption of the device.

4.2.1 802.11g Specification

IEEE 802.11g is an amendment to the IEEE 802.11 specification that extends throughput up to 54 Mbit/s using a 2.4 GHz band (same as 802.11 and 802.11b, thus this second one with a throughput of 11 Mbit/s). This specification has received the marketing name of Wi-Fi all over the world and it is the most used Wireless LAN standard nowadays.

The main characteristics of this WLAN specification are the following.

- Released on June 2003
- Frequency: 2.4 GHz
- Maximum bitrate: 54 Mbit/s

- Modulation: OFDM

802.11g was the third standard for wireless networks, after 802.11a and 802.11b (both from 1999). The main differences are that 802.11a operates with a frequency of 5 GHz and 802.11b has a maximum bitrate of only 11 Mbit/s. Since 802.11b and 802.11g use the same frequency, hardware for both kind of networks is compatible, but the inclusion of a 802.11b card in a 802.11g network will slow down the whole traffic within the WLAN.

The major disadvantage of a 802.11g network is the fact that it suffers from the interferences occurred in the 2.4 GHz frequency range, as well as the 802.11b. The usual devices that cause this interferences in the same frequency range are Bluetooth connections, microwave ovens, baby monitors and some cordless telephones.

4.2.2 Power saving methods

Now focusing on the power consumption of a WLAN device, the first important thing to talk about is the protocol modes. These protocol modes are defined in WLAN (and also Bluetooth) standards, although the way they are implemented may vary the power consumption in one way or another. The modes are defined by combinations of different states that the physical layers can take. Those states are the following.

- **Off.** The only power consumption is leakage current, but coming out of the off state can take a long time (many milliseconds).
- **Sleep/Standby.** The device may be consuming as little as 0.175 mW and can wake quickly unless the main crystal is turned off.
- **Listen.** The device is listening for a packet to arrive, so most of the radio receiver must be on. State-of-the-art power numbers for WLAN and Bluetooth devices in this mode are 110 and 46 mW, respectively.
- **Active Rx.** Similar to the listen state, but use of additional circuitry may push power consumption for WLAN (802.11g) and Bluetooth devices to 140 and 52 mW, respectively.
- **Active Tx.** In the transmit state, the device's active components include the RF power amplifier, which often dominates in high-power transmit systems. State-of-the-art power consumption for an 802.11g WLAN device is 450 mW at 15 dBm Tx power. For Bluetooth, values are 78 mW at 3 dBm Tx power and 55 mW at "18 dBm.

Therefore, using combinations of these states, different protocol modes can be created, such as:

- Searching for a network
- Connected but idle
- Media traffic flow
- Max-throughput traffic flow

With the N810 power save mode option activated, the device spends most of the time in the “Sleep” mode, even while receiving and sending information.

4.2.3 Beacon Frames

The beacon frames are management frames that enables stations in a wireless LAN to establish and maintain communications. They play a major role in the power consumption of a wireless LAN card, since they control the sleep mode of the HW.

This beacon frames have nothing to do with data frames from higher layers, they are somehow the “heartbeat” of a wireless LAN, giving information about the network to all the stations periodically.

A beacon frame has typically a fifty bytes length, and half of those are headers and CRC (the most usual method used in order to detect transmission errors in the frames). The header includes the destination address (always broadcast address, since the destination are all the stations within the network).

The body of the beacon frame is located between the header and the CRC. The following are the different fields included in the beacon frame body:

- **Beacon interval.** This interval is the amount of time between two beacon frame transmissions. The stations must know this time in order to wake up after having entered power save mode. If the stations do not know this time, they would not be able to retrieve the beacon frames and maintain the connection to the wireless network.
- **Timestamp.** The stations use this timestamp to update their local clocks. With this, all the stations related with the same WLAN access point will be properly syncrhonized.

- **SSID.** It is the identifier of a specific wireless LAN. Access points include the SSID in the beacon frame to enable sniffing functions or automatically configure the NIC (network interface card), but some access points have the option to disable this SSID field in the beacon frame because of security reasons.

- **Supported rates.** Each beacon frame includes the supported transmission rates in the own network, so the stations can use performance metrics to decide which access point to associate with.

- **Parameter sets.** This includes information about signaling methods, such as frequency hopping spread spectrum, direct sequence spread spectrum, etc.) and the channel number used by the access point.

- **Capability information.** This field consists on requirements that the stations must satisfy in order to belong to the WLAN. E.g. the encryption algorithm used in the network.

- **TIM.** Acronym for "Traffic Indication Map", this identifies the stations that are currently using power saving mode and have data frames waiting for them in the access point's buffer.

4.3 Measurement Setup

This section explains the procedure followed to carry out the measurements.

First of all we will explain the wireless access point settings. Some of these settings change in the different use cases, but the "standard" setup is the following:

- Transmission rate: 54 Mbps
- CTS protection mode: Disabled
- Beacon interval: 100 ms
- DTIM interval: 1 ms
- Fragmentation threshold: 2346 bytes
- RTS threshold: 2347 bytes
- AP isolation: Off

- RF channel: 11

Among the values listed above, the transmission rate and the beacon interval will change in certain use cases. The remaining parameters are constant for all use cases.

In order to know what behaviour the mobile device has, apart from the file transfers in which the use cases consist of, there are some other settings affecting the power consumption. They are listed below.

- Bluetooth: Disabled
- Blacklight sensor: Minimum
- Speaker volume: Muted
- Screen power save: On
- WLAN power save: On

Figure 4.1 shows how the mobile device and the multimeter are connected.

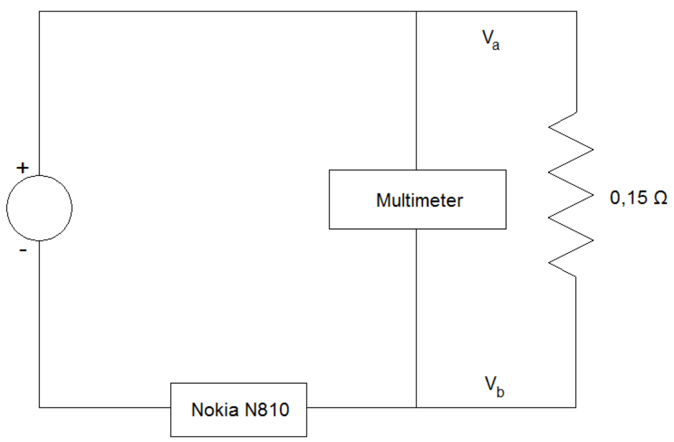


Figure 4.1: Measurement station.

The power supply always provides a constant voltage of 3.9 Volts, and the resistance is the same for all use cases, R = 0.15 Ohm. Using Ohm's Law, the current across the resistance is calculated.

$$I = \frac{V_b}{R} \tag{4.1}$$

Once the current is calculated, we can easily obtain the power consumption.

$$P = (V_a - V_b)\, I = \frac{(V_a - V_b)\, V_b}{R} \tag{4.2}$$

Now that we have the power calculated, the energy is obtained with:

$$E = Pt \tag{4.3}$$

Where “t” is time, but not the total amount of time spent in the file transfers, but the intervals of constant voltage during the transfers, provided by the FlukeView Forms software.

4.4 Power Measurement Use Cases

In this section we will mainly explain the uses cases of the study, which will be analyzed in the following chapter.

There are 10 use cases in total, although each one includes more than only one measurement, in order to establish relations between the parameters that vary and the ones that keep constant in each case.

The design of the use cases tries to cover all the possible factors within a wireless network and therefore determine those that have more influence on the power consumption.

To carry out the analysis, different files will be sent and received, varying different parameters either from the network or from the files themselves, except for the last use cases, in which we will measure direct instant messages, which will have a different behaviour, especially when sending them from the N810, since tapping on the screen or writing with the keyboard produces a higher power consumption than just accepting the delivery or receipt of a file.

All these factors are the following:

- **Size of the files sended or received.** With this, we will try to establish a relation between the amount of data transfered and the power consumption, considering the transmission speed.

- **Type of the files sended or received.** Since the MSNP protocol may have a different treatment depending on the file type, this may have some kind of repercussion on the power consumption.
- **Beacon interval of the WLAN.** The beacon interval has a very important role on the activity of any wireless device. The lower the beacon interval, the higher the activity that the device will perform in order to communicate with the wireless network, therefore, there will be variations on the power consumption.
- **Transfer rate of the WLAN.** Faster communications may produce smaller transfer times, but the reduction on this time may affect the power increasing its consumption. Here we will try to establish a relation between the transfer rate and the power.
- **Peer locations of the message sender or receiver.** Since the N810 is always connected to a wireless network, we will try to find out whether the location of the other agent of the communication (same or different network, wired or wireless) has a consequence on the final power consumption.
- **Frequency and length of text messages sended or received.** In order to analyze this factor, the messages sended and received will be instant text messages instead of files. Tapping on the screen or writing several characters probably has a major role on the power consumption.

For each use case, the factors that remain constant are shown in this chapter. Those other factors that vary can be seen in the results tables in the next chapter.

4.4.1 Case 1. File Size and Type (Receiving)

This first use case is the most basic of them all. The procedure consists on receiving files with different sizes and types from a terminal to the N810 tablet. All the wireless network settings are kept constant, so the only variation is the size and type of the files.

These are the factors that remain constant in this use case:

- Peer locations: Both N810 and second client connected to the same wireless network.

- WLAN transmission rate: 54 Mbps
- WLAN beacon interval: 100 ms

Given this scenario, the N810 will receive 5 different file types, each one with different sizes. They are listed in table 4.1.

File type	**File size (KB)**
Text	256
Text	512
Text	1024
Text	2048
Text	4096
Image (.jpg)	628
Image (.jpg)	1040
Image (.jpg)	1987
Image (.jpg)	5698
Image (.bmp)	256
Image (.bmp)	514
Image (.bmp)	1028
Image (.bmp)	2013
Audio (.mp3)	1029
Audio (.mp3)	2110
Audio (.mp3)	4322
Audio (.mp3)	8440
Audio (.ogg)	1307
Audio (.ogg)	2337
Audio (.ogg)	4210
Audio (.ogg)	11689

Table 4.1: File types and sizes for use case 1.

4.4.2 Case 2. File Size and Type (Sending)

This case is analogous to the previous one, and this will happen from now on for all the remaining use cases. Given a use case with specific settings, the next one will have the same settings, except for the transmission of the files, that will be the opposite to the previous case. I.e. if the first case consisted on receiving files, the next one will deal with sending the files.

Therefore, the constant settings for this second use case are the same that those for the first one:

- Peer locations: Both N810 and second client connected to the same wireless network.
- WLAN transmission rate: 54 Mbps
- WLAN beacon interval: 100 ms

The different files used are shown in table 4.2.

File type	File size (KB)
Text	256
Text	512
Text	1024
Text	2048
Text	4096
Image (.jpg)	628
Image (.jpg)	1040
Image (.jpg)	1987
Image (.jpg)	5698
Image (.bmp)	256
Image (.bmp)	514
Image (.bmp)	1028
Image (.bmp)	2013
Audio (.mp3)	1029
Audio (.mp3)	2110
Audio (.mp3)	4322
Audio (.mp3)	8440
Audio (.ogg)	1307
Audio (.ogg)	2337
Audio (.ogg)	4210
Audio (.ogg)	11689

Table 4.2: File types and sizes for use case 2.

4.4.3 Case 3. WLAN Beacon Interval (Receiving)

In this third case, the parameter that will vary will be the WLAN beacon interval, while we send a file from a terminal to the N810. This beacon interval determines the amount of time between one beacon frame transmission and the next one.

The constant settings for this use case are the following:

- Peer locations: Both N810 and second client connected to the same wireless network.
- WLAN transmission rate: 54 Mbps
- File type: Plain text
- File size: 1024 KB

The values taken by the beacon interval are shown in table 4.3.

Beacon interval (ms)
1
20
50
100
200
500

Table 4.3: Beacon interval values for use case 3.

4.4.4 Case 4. WLAN Beacon Interval (Sending)

This use case is analogue to the previous one. Here, the file will be sent from the N810 to the other terminal. As expected, the constant and varying parameters are the same that in the previous case.

- Peer locations: Both N810 and second client connected to the same wireless network.
- WLAN transmission rate: 54 Mbps
- File type: Plain text
- File size: 1024 KB

Table 4.4 shows the values for the beacon interval in this case.

Beacon interval (ms)
1
20
50
100
200
500

Table 4.4: Beacon interval values for use case 4.

4.4.5 Case 5. WLAN Transfer Rate (Receiving)

The next analyzed parameter will be the WLAN transfer rate. In this use case, again we will send a file from the different terminal to the N810.

A higher transfer rate may increase the transmission speed, but it will also mean a higher activity of the stations in the network, since the package processing will have to be faster if the packages arrive more frequently.

The parameters that will not vary in this case will be the following:

- Peer locations: Both N810 and second client connected to the same wireless network.
- WLAN beacon interval: 100 ms
- File type: Plain text
- File size: 1024 KB

The different values of the transmission rate are shown in table 4.5.

WLAN Transmission Rate (Mbps)
1
6
12
24
54

Table 4.5: WLAN transmission rate values for use case 5.

4.4.6 Case 6. WLAN Transfer Rate (Sending)

Again, this case is similar to the previous one, but here the file will be sent from the N810 to the other terminal in the network. The constant settings are the same:

- Peer locations: Both N810 and second client connected to the same wireless network.
- WLAN beacon interval: 100 ms
- File type: Plain text
- File size: 1024 KB

Again, the transmission rate values are shown in table 4.6.

WLAN Transmission Rate (Mbps)
1
6
12
24
54

Table 4.6: WLAN transmission rate values for use case 6.

4.4.7 Case 7. Different Peer Locations (Receiving)

With this use case, we will try to explain if the location of one of the clients involved in the communication has influence on the power consumption of the other client.

This may have some influence on the activity of the N810, since if both clients are located in the same network, this will increase the packet traffic within this network, and therefore, the stations will have to process more packets, what will cause a higher activity (and presumably higher power consumption as well).

The constant parameters in this case are the following:

- WLAN beacon interval: 100 ms
- WLAN transmission rate: 54 Mbps

- File type: Plain text
- File size: 1024 KB

The different locations used for the second client are shown in table 4.7.

Location of the second IM client
Same network as the first one, wireless-connected
Same network as the first one, wired-connected
Different network than the first one

Table 4.7: Locations of the second IM client in use case 7.

When using the same network, only the two terminals having the clients will be connected in the network (as it happened in all the previous cases), in order to avoid an excessive traffic.

For the second location, the terminal having the second client will be connected to the same router that provides the wireless network, but via a network cable.

For the last case, the terminal having the second client will be connected via network cable to a different network than the WLAN with the N810.

4.4.8 Case 8. Different Peer Locations (Sending)

Once again, this case contains the same settings as the previous one, but the file is sent from the N810 to the other terminal.

The following are the constant parameters:

- WLAN beacon interval: 100 ms
- WLAN transmission rate: 54 Mbps
- File type: Plain text
- File size: 1024 KB

Table 4.8 shows the different locations of the second IM client.

Location of the second IM client
Same network as the first one, wireless-connected
Same network as the first one, wired-connected
Different network than the first one

Table 4.8: Locations of the second IM client in use case 8.

4.4.9 Case 9. Instant Messages (Receiving)

These two last cases are slightly different from all the previous ones. Here we will analyze the power consumption based in receiving instant messages in the N810, of course sent from the second terminal used before. Therefore, there will not be a file transfer, but only short messages written character by character (or several characters at the same time, by copying and pasting a string of characters).

With this new scenario, we will try to extract some conclusions about the total amount of characters sent and the frequency of the messages, writing them with different lenght and waiting different time intervals between a message and the next one.

The settings for this use case are the following:

- Peer locations: Both N810 and second client connected to the same wireless network.
- WLAN beacon interval: 100 ms
- WLAN transmission rate: 54 Mbps
- File type: Plain text
- File size: 1024 KB

Tables 4.9 and 4.10 show the different amount of characters sent and the frequency of the messages.

Total amount of characters sent per message
10
20
50

Table 4.9: Characters per message in use case 9.

Time interval between messages (s)
1
5
10
20

Table 4.10: Time elapsed between messages in use case 9.

Therefore, there will be twelve different tests for this use case.

When sending the messages from the terminal to the N810, all of them will be sent having the string with the total amount of characters copied, and pasting them at once (since obviously it is impossible to write 50 characters in one second typing them one by one). This has no meaning on the power consumption, since all the sizes (10, 20 and 50) are encapsulated in one only packet of the MSN protocol, so the power consumption differences due to the packet size are minimal. It would be different if the messages had to be sent in more than one packet.

4.4.10 Case 10. Instant Messages (Sending)

This last case also deals with instant messaging instead of file transfers, as well as the previous one. Now the messages will be sent from the N810 to the second client.

The WLAN settings will remain constant again, and the varying parameters will be the length and frequency of the messages sent.

Later, we will see that clicking on the keyboard or tapping in the tactile screen of the N810 causes a great increase on the power consumption, especially if the tablet had the display light off. In order to know how to establish a relation between the clicking and tapping (in fact, writing messages) and the power consumption, the strings sent will be written typed character by character, using different lenghts and frequencies, as said before.

Again, the network settings will remain constant:

- Peer locations: Both N810 and second client connected to the same wireless network.
- WLAN beacon interval: 100 ms
- WLAN transmission rate: 54 Mbps

- File type: Plain text
- File size: 1024 KB

Once again, the length and frequency of the messages sent are shown in tables 4.11 and 4.12.

Total amount of characters sent per message
5
10
20

Table 4.11: Characters per message in use case 9.

Time interval between messages (s)
5
10
20
30

Table 4.12: Time elapsed between messages in use case 10.

When the messages are sent from the N810 to the other terminal, the messages will be written character by character, in order to see how each click on the screen or key pressed increases the power consumption.

Chapter 5

EXPERIMENTAL RESULTS AND ANALYSIS

This chapter presents the results of the measurements described in the previous one, shown in tables and charts, in order to understand the power consumption behaviour in a quicker way. These charts will show the total amount of power consumption depending on the different factors that vary with each use case, like the file size or the WLAN beacon interval, but sometimes it will be also relevant to take a look to the total time used in the transference or the average speed.

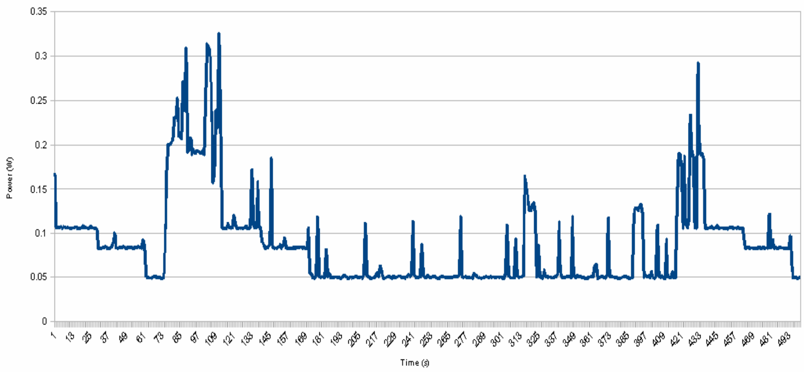


Figure 5.1: Overall power consumption in a typical connection.

The chart above (figure 5.1) presents the power consumption during a short IM session. At the beginning, the figure shows a consumption of around 0.11W (device idle but with the light on). When the light switches off the consumption is reduced to 0.08W and when the screen switches off completely the consumption is 0.05W. With this few data we can have an idea that the screen light causes a considerable increase in the overall consumption.

Now taking a look on the rest of the chart, which concerns the IM, we can see a first great increase, between 0.16W and 0.33W, which corresponds with the logging in to the IM system. Once logged in, there are several ping signals which add around 0.06W when they are received. Later on, a contact gets connected and disconnected, with a consumption 0.12W and 0.08W respectively. Finally, the last higher increase corresponds with the disconnection of our user, which reaches 0.29W of consumption. It is important to note again that, every time we tap on the screen to perform the required actions to log off, the screen light switches on, and also a simple tap makes the power consumption rise considerably.

Next sections will cover a deeper analysis of the use cases specified in the previous chapter.

5.1 Case 1. File Size and Type (Receiving)

From now on, each section covers the results of every use case described in the previous chapter.

This first one consists on sending files with different sizes and types to the N810 from a different terminal. The results are summarized in the following table, which shows the type of the files used in the transfers, their size in KB, the total amount of time elapsed in each transfer, the power and energy consumption and the average speed of the file delivery.

Type	Size (KB)	Time (s)	Power (mW)	Energy (J)	Speed (KB/s)
Text	256.00	69.20	165.70	92.32	3.70
Text	512.00	127.40	238.74	138.31	4.02
Text	1024.00	206.90	363.68	200.88	4.95
Text	2048.00	371.10	610.53	319.24	5.52
Text	4096.00	755.00	1189.33	603.78	5.43
Image (.jpg)	628.37	124.70	270.77	136.76	5.04
Image (.jpg)	1040.21	195.70	352.83	187.42	5.32
Image (.jpg)	1987.01	366.10	607.36	317.08	5.43
Image (.jpg)	5698.61	976.70	1750.97	819.44	5.83
Image (.bmp)	256.05	71.40	179.92	92.57	3.59
Image (.bmp)	514.78	121.70	258.37	133.99	4.23
Image (.bmp)	1028.06	210.70	370.56	202.09	4.88
Image (.bmp)	2013.82	382.90	644.20	333.31	5.26
Audio (.mp3)	1029.49	211.80	410.00	217.00	4.86
Audio (.mp3)	2110.18	374.00	768.80	389.89	5.64
Audio (.mp3)	4322.76	761.40	1474.67	701.33	5.68
Audio (.mp3)	8440.71	1496.50	2833.39	1369.61	5.64
Audio (.ogg)	1307.72	252.90	484.66	261.88	5.17
Audio (.ogg)	2337.87	442.10	894.59	429.93	5.29
Audio (.ogg)	4120.63	707.30	1414.58	671.95	5.83
Audio (.ogg)	11689.37	1748.40	3180.90	1635.94	6.69

Table 5.1: Measurement results in case 1.

As it can be seen in table 5.1, the bigger the file sent is, the higher the consumption, but this is mainly caused due to the higher amount of time needed to finish the delivery. The next chart shows the relation between the size of the file and the total consumption, for all file types.

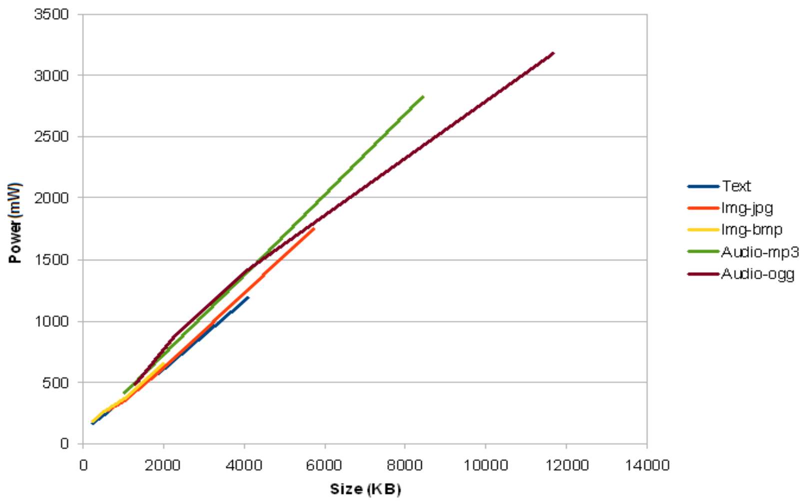


Figure 5.2: Power consumption depending on file type and size when receiving files.

The relation shown in figure 5.2 is practically linear between the size and the consumption, and there are very few differences in the consumption depending on the file types. This makes sense since, as explained in chapter 4, the files are sent embedded in MSG packets, therefore, the file type and its internal organization is completely irrelevant when it comes to perform a file transfer.

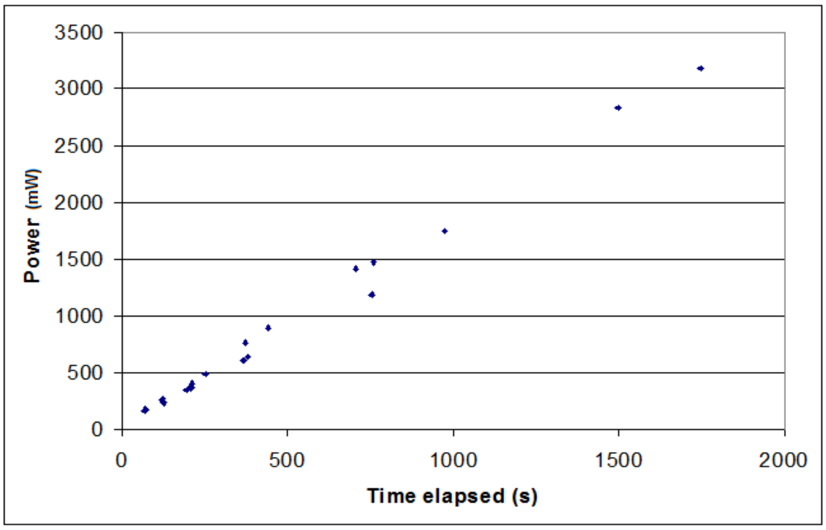


Figure 5.3: Power consumption depending on time elapsed when receiving files.

Unlike the type, the size of the file is relevant, but in the sense that bigger files require more time to finish the transfer, so the main factor affecting the consumption is the total time of the transfer, since the consumption is more or less constant during all the time the device is performing the necessary activities to carry out the delivery. Figure 5.3 shows the consumption depending on the total time of the transfer. As it could be expected, the consumption follows a straight line, showing a linear dependency on the time elapsed, although not perfect, due to minimal differences on the transfer speeds.

5.2 Case 2. File Size and Type (Sending)

This is the analogue case to the previous one. Files with different sizes and types are sent, but this time, from the N810 to a different terminal. Results are shown in table 5.2.

Type	Size (KB)	Time (s)	Power (mW)	Energy (J)	Speed (KB/s)
Text	256.00	69.20	147.96	115.23	3.70
Text	512.00	138.80	296.34	153.79	3.69
Text	1024.00	244.90	491.96	247.73	4.18
Text	2048.00	534.00	956.92	475.38	3.84
Text	4096.00	1042.60	1754.64	885.57	3.93
Image (.jpg)	628.37	153.00	321.27	167.80	4.11
Image (.jpg)	1040.21	273.10	499.23	263.97	3.81
Image (.jpg)	1987.01	499.40	887.02	447.13	3.98
Image (.jpg)	5698.61	1494.60	2587.43	1235.06	3.81
Image (.bmp)	256.05	65.60	159.40	95.58	3.90
Image (.bmp)	514.78	133.50	254.50	150.80	3.86
Image (.bmp)	1028.06	292.70	503.87	259.86	3.51
Image (.bmp)	2013.82	534.20	885.45	463.84	3.78
Audio (.mp3)	1029.49	293.80	502.88	262.93	3.50
Audio (.mp3)	2110.18	550.20	897.28	472.42	3.84
Audio (.mp3)	4322.76	1018.60	1701.57	878.22	4.24
Audio (.mp3)	8440.71	2171.60	3723.49	1795.00	3.89
Audio (.ogg)	1307.72	329.70	572.95	306.47	3.97
Audio (.ogg)	2337.87	635.40	1148.92	555.92	3.68
Audio (.ogg)	4120.63	1085.60	1770.79	901.77	3.80
Audio (.ogg)	11689.37	2667.10	4362.47	2149.66	4.38

Table 5.2: Measurement results in case 2.

The results in this case are pretty similar to the previous one, except for the fact that the consumption is notably higher in this case. The highest consumption in case 1 was 3180.90 mW, for the 11 MB .ogg file, and in this case, the same file requires 4362.47 mW. Next charts show the consumption depending on both the size of the files and the time elapsed.

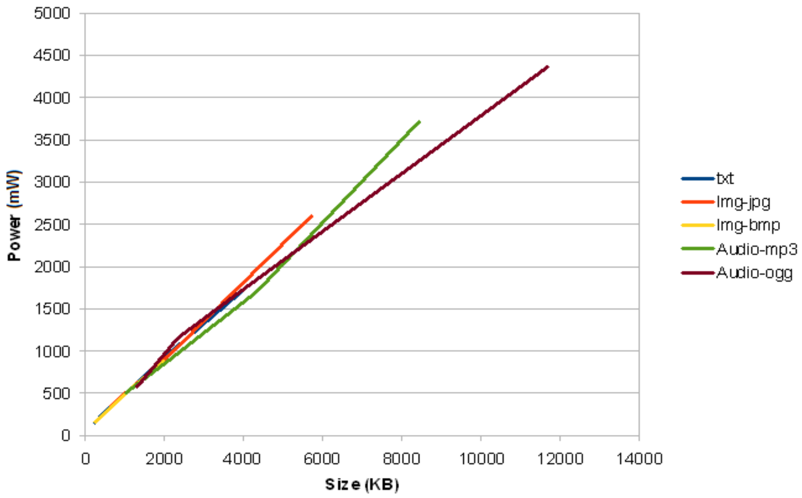


Figure 5.4: Power consumption depending on file type and size when sending files.

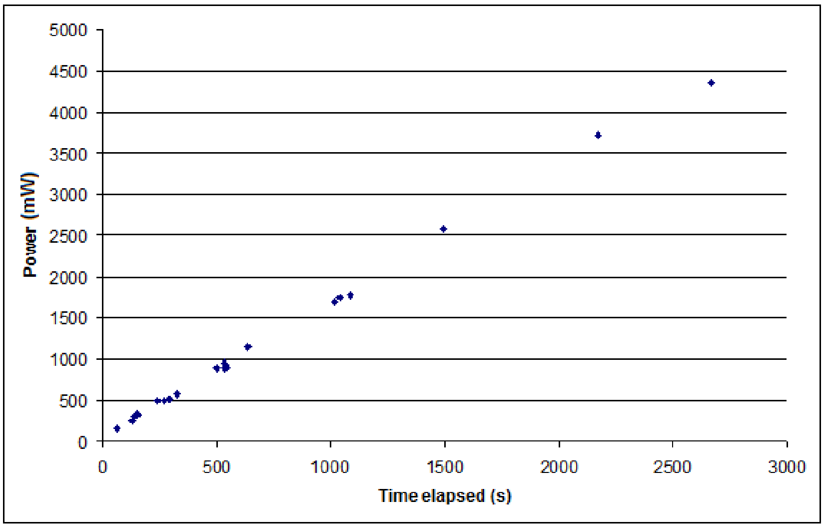


Figure 5.5: Power consumption depending on time elapsed when sending files.

In both figures 5.4 and 5.5 the linear dependency can be seen again, with, as said before, the only difference that the consumption is higher in this case, when the N810 is sending the files instead of receiving them. Obviously, the file type is irrelevant also in this case.

In the next chart (figure 5.6) both values from cases 1 and 2 are compared, in order to view graphically the higher consumption in case 2. In general, the average consumption when sending a file from the N810 is 26% higher than when receiving it.

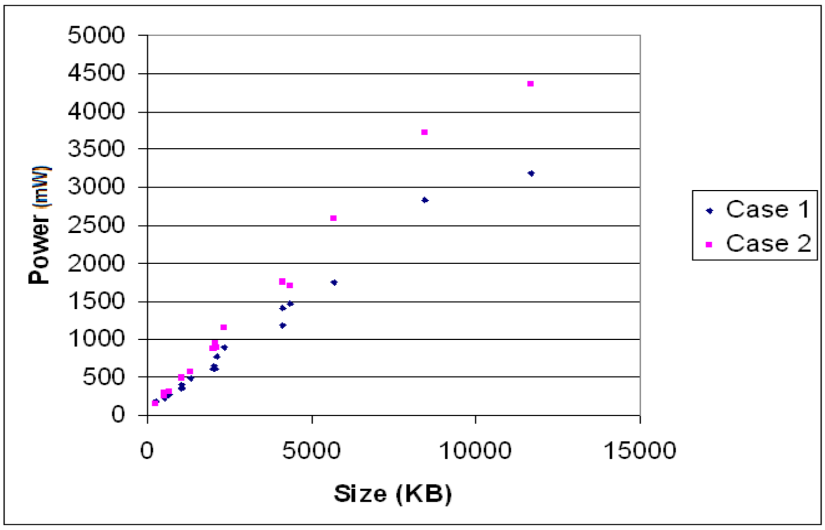


Figure 5.6: Comparison between the consumption in cases 1 and 2.

5.3 Case 3. WLAN Beacon Interval (Receiving)

In this case, the file is the same in all transfers and the variation occurs in the WLAN beacon interval. The files are received in the N810.

Beacon (ms)	Time (s)	Power (mW)	Energy (J)	Speed (KB/s)
1	461.70	775.63	601.65	2.22
20	461.70	428.67	277.75	2.22
50	458.20	407.54	253.26	2.23
100	460.80	389.03	248.54	2.22
200	463.50	393.20	247.66	2.21
500	467.70	425.58	250.17	2.19

Table 5.3: Measurement results in case 3.

In this case, the times and average speeds are very similar, so this time we can analyze how the consumption is affected by the beacon interval regardless the time elapsed. As shown in figure 5.7, the first case (1 ms beacon interval) presents the highest consumption compared to any other case, and the time and speed are similar to the other ones.

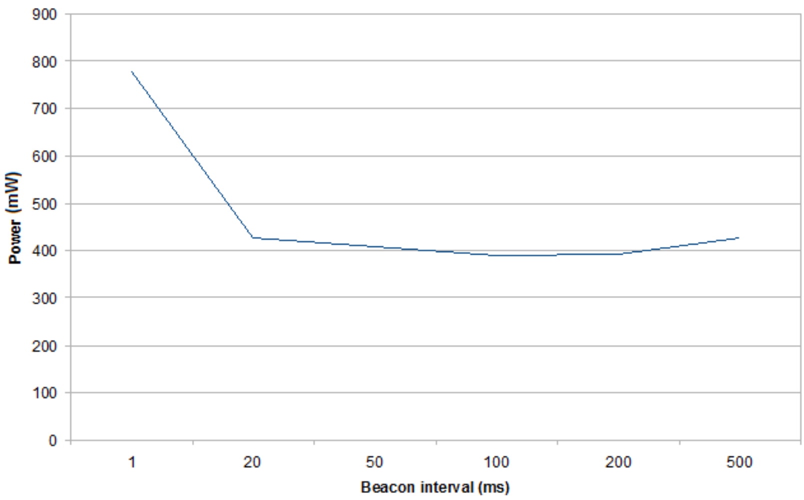


Figure 5.7: Power consumption depending on WLAN beacon when receiving files.

As it could be expected, with the extremely low beacon interval of 1 ms, the device must carry out much more activity in order to check the WLAN beacon frames, which greatly increases the power consumption. With other settings than 1 ms, the consumption is more or less the same, and the minimal differences are due to the different times of the transfers.

5.4 Case 4. WLAN Beacon Interval (Sending)

Again, the WLAN beacon interval varies while the file size and type remains constant. This time the files are sent from the N810.

Beacon (ms)	Time (s)	Power (mW)	Energy (J)	Speed (KB/s)
1	261.70	583.31	420.75	3.91
20	212.90	452.56	247.75	4.81
50	242.90	465.76	246.83	4.22
100	278.80	504.85	257.98	3.67
200	361.90	579.51	295.00	2.83
500	443.10	689.84	332.53	2.31

Table 5.4: Measurement results in case 4.

This time the beacon interval has a greater influence than in the previous case, in which the device that sent the files was connected to a wired network, but now, the beacon not only affects the power consumption, but also the time and speed.

Figure 5.8 shows the relation between power and beacon interval, like in the previous case.

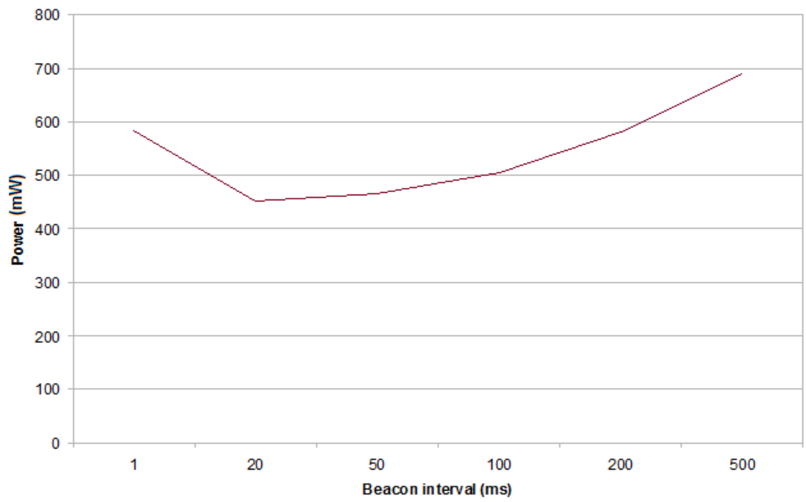


Figure 5.8: Power consumption depending on WLAN beacon when sending files.

The differences are obvious. While in the previous case the only different setting was the one with 1 ms, here, any variation on the beacon interval causes a different consumption. Again, with the beacon interval set to 1 ms, the device must perform a higher activity to chech the beacon frames, which cases a high consumption, but except for that one, the more the beacon interval increases, the more the consumption does, but in this case, the device must wait more and more time to check the frames and therefore, to interact with the network, what makes the transfer time higher and higher, and therefore the consumption.

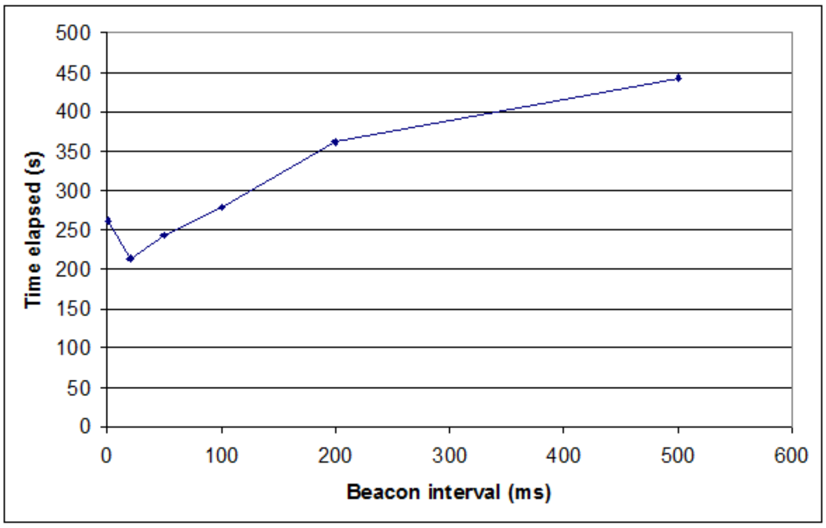


Figure 5.9: Time elapsed depending on WLAN beacon when sending files.

This second chart (figure 5.9) shows the time elapsed on the file delivery depending on the beacon interval. To summarize, a very low beacon interval causes a high consumption due to the high activity the device must perform, but a very high beacon interval reduces the interaction with the network, causing the speed to slow down and therefore, the time to increase, which is the factor that most affects the power. As seen in the chart, the best beacon interval setting is 20 ms, although the default in most wireless networks is 100 ms.

5.5 Case 5. WLAN Transfer Rate (Receiving)

Now the parameter that varies is the WLAN transfer rate. The files are received in the N810.

Rate (Mbps)	Time (s)	Power (mW)	Energy (J)	Speed (KB/s)
1	465.70	397.09	257.68	2.20
6	461.80	384.40	248.77	2.22
12	458.60	379.74	250.52	2.23
24	463.00	379.53	247.63	2.21
54	454.90	391.14	245.28	2.25

Table 5.5: Measurement results in case 5.

The differences due to the variations in the WLAN transfer rate are minimal. There is no more than 20 mW between the highest and the lowest consumption, and with higher transfer rates, lower transfer times. However, the last case, set to 54 Mbps, surprisingly presents a consumption higher than expected, having the lowest transfer time.

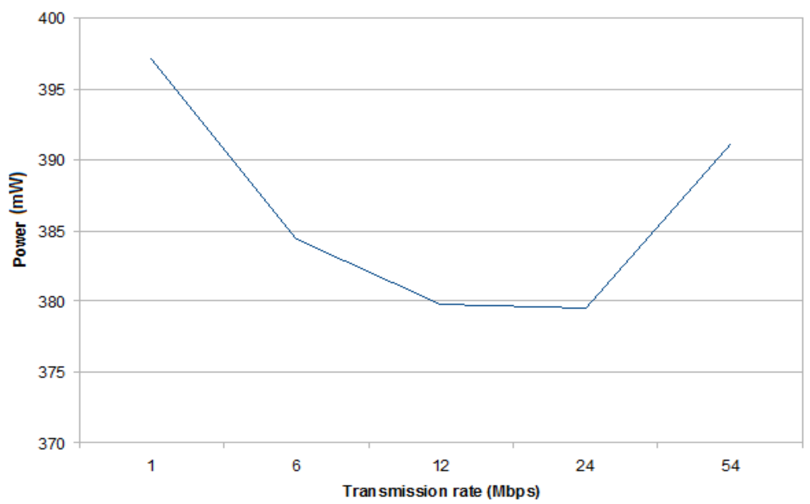


Figure 5.10: Power consumption depending on WLAN transmission rate when receiving files.

Due to unknown reasons, the consumption raises in the last case, instead of keep going down, as could be expected, but anyway, the variation between transfer rate and time elapsed is not uniform, so it is difficult to analyze whether the transfer rate affects greatly or not the time or power consumption. However, since the difference between the highest and the lowest consumption is less than 20 mW, it can be said that the WLAN transfer rate does not affect greatly the consumption (but this does not mean that it does not affect it at all), at least not as much as the WLAN beacon interval.

5.6 Case 6. WLAN Transfer Rate (Sending)

Again, the WLAN transmission rate varies, but this time the file is sent from the N810.

Rate (Mbps)	Time (s)	Power (mW)	Energy (J)	Speed (KB/s)
1	256.10	477.59	251.94	4.00
6	252.30	480.74	248.13	4.06
12	269.30	505.27	259.68	3.80
24	271.10	534.41	263.74	3.78
54	245.90	471.03	249.05	4.16

Table 5.6: Measurement results in case 6.

Once again, the results are a little tricky, since the average speeds obtained are different among the use cases.

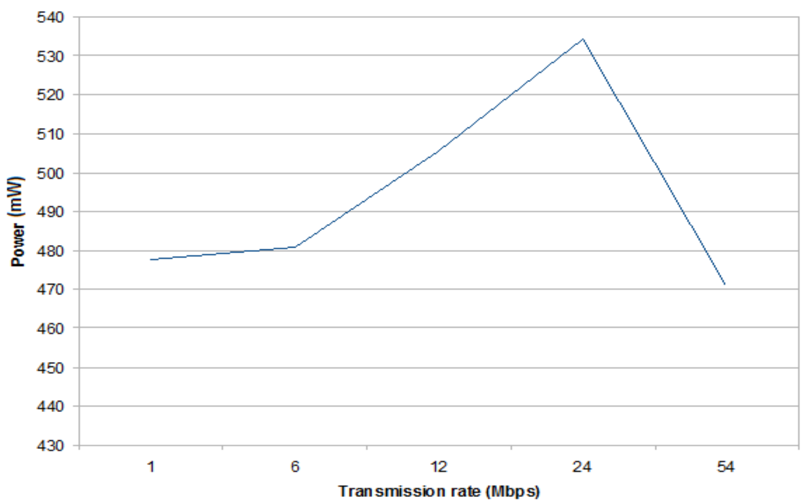


Figure 5.11: Power consumption depending on WLAN transmission rate when sending files.

This time, the highest transfer rate causes the fastest transfer, and therefore the lowest consumption, but for example, the cases with 12 and 24 Mbps are strange, since their transfer times are not very different (less than 2 s), but the consumptions are (almost 30 mW).

Figure 5.11 shows the strange results obtained, especially the third and fourth case, presumably affected by uncontrolled conditions in the network (packet loss, noise, etc.). The remaining cases (1, 6 and 54 Mbps) show a logical behaviour, with lower transfer times when the rate is higher, but the difference in consumption is again very small (less than 10 mW), so, as said with the previous use case, the transfer rate is a factor that affects minimally the power consumption.

5.7 Case 7. Different Peer Locations (Receiving)

Now the variation occurs in the second client's location, being the same network (with the same or different access point than the N810) or a different network. The N810 receives the file.

Locations	Time (s)	Power (mW)	Energy (J)	Speed (KB/s)
Different networks	470.50	421.61	253.59	2.18
Same network, diff. access point	464.60	400.54	246.45	2.20
Same network and access point	472.60	397.72	256.68	2.17

Table 5.7: Measurement results in case 7.

Although there is no way to control the location of the partners with whom we are having an IM session, the results show that there are differences between this use case settings.

The configuration that causes a higher consumption is that in which the second client is located in a different network than the N810, mainly due to the routing.

If the transfer is carried out in the same network, the consumption is lower, regardless the time used, since as seen in the results, the time of the third case is higher than the first one (being in different networks), but the consumption is lower.

As written in table 5.7, figure 5.12 shows that the higher consumption is obtained when the devices are located in different networks. When located in the same network, the consumption is lower if both devices use the same access point (even with a high transfer time obtained in the experiment). In this case, both clients were using the wireless connection.

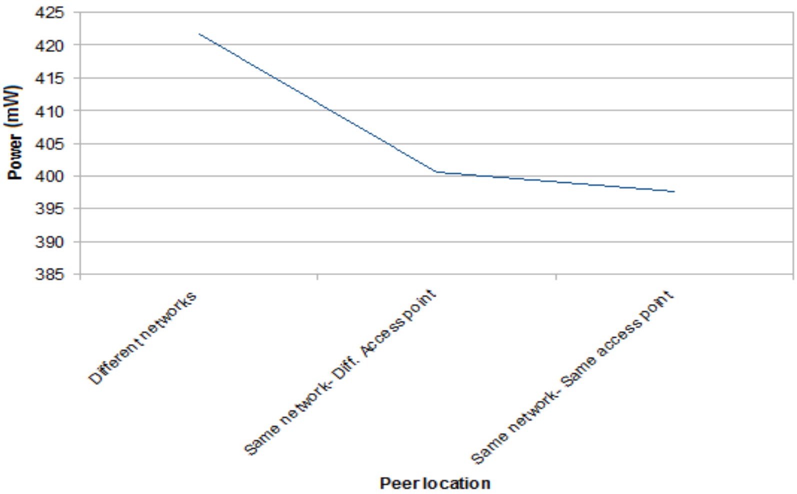


Figure 5.12: Power consumption depending on peer locations when receiving files.

5.8 Case 8. Different Peer Locations (Sending)

Again, the peer location of the second client varies. This time the N810 sends the file to this second client.

Locations	Time (s)	Power (mW)	Energy (J)	Speed (KB/s)
Different networks	264.80	482.13	254.47	3.87
Same network, diff. access point	264.10	486.34	251.71	3.88
Same network and access point	232.80	448.95	239.59	4.40

Table 5.8: Measurement results in case 8.

In this case there is the same problem as in the use cases 5 and 6, where the results are not completely clear. The main point is that when both devices are connected to the same network and access point, the transfer is faster and therefore the consumption lower, but in the other cases, it is strange the fact that being in different networks, the consumption is lower than being in the same network with different access points, with a higher time in addition.

However, the differences are minimal (less than 1 s in time and than 4 mW in power).

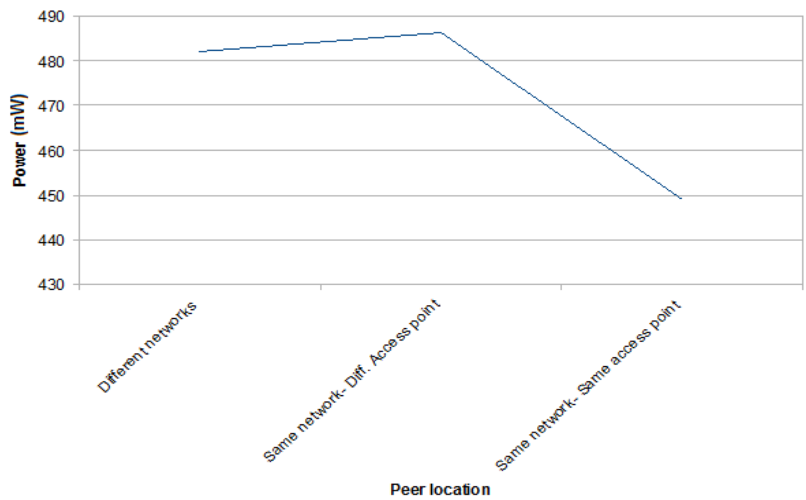


Figure 5.13: Power consumption depending on peer locations when sending files.

Figure 5.13 shows the results presented in the table. With both devices connected to the same network and access point, the consumption is slightly lower than in other settings, either sending or receiving the files. With other configurations, it is hard to establish which behaviour is better, since it is different when sending than when receiving, but anyway, outside the experiments, focusing on any IM session, it is impossible to force both clients to be located in the same access point, due to the fact that this depends on the physical location of the people using the software.

5.9 Case 9. Instant Messages (Receiving)

For this use case, several instant messages with different amounts of characters and times between each message are sent to the N810.

Chars	Interval (s)	Chars/s	Time (s)	Power (mW)	Energy (J)
10	1	10	87.80	160.97	83.85
10	5	2	83.30	103.53	69.34
10	10	1	83.20	79.42	51.96
10	20	0.5	79.60	56.49	41.06
20	1	20	79.80	158.55	81.68
20	5	4	73.60	97.98	64.63
20	10	2	74.80	74.30	48.55
20	20	1	80.60	52.71	39.07
50	1	50	83.70	157.78	83.24
50	5	10	72.70	88.35	63.69
50	10	5	78.50	72.37	49.35
50	20	2.5	72.90	46.66	33.96

Table 5.9: Measurement results in case 9.

This is the most interesting case from the user's point of view, since IM software is most used to send text messages.

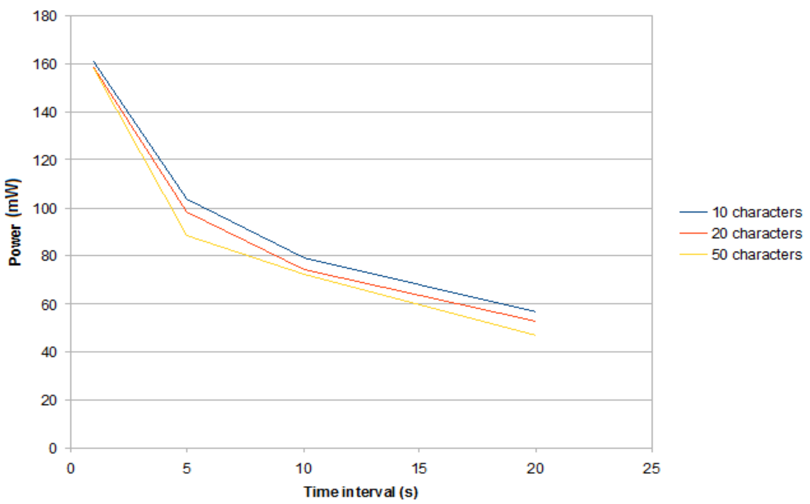


Figure 5.14: Power consumption depending on time interval between received messages.

Here, the average speed is not shown, because it is more or less irrelevant due to the aspects that are analyzed. The tests in this case have consisted on sending messages with the lenght and frequency specified in the table, but the time shown above is the total amount of time that FlukeView Forms recorded on each measurement, stopped when the activity ceased in the N810. To have a clearer analysis of this scenario is better to take a look to the charts.

Figure 5.14 shows the consumption depending on the time interval between each message. As can be seen, the lenght of the message affects very little the power consumption since, with so small texts, the whole message is kept in a single MSG packet, so the variation is only a few bytes. Unlike the lenght of the message, the time between them is very important. The chart shows how the consumption is reduced if the time between messages increases. High waiting times allow the N810 to reduce its activity, and switch off its screen light, what helps a lot when it comes to save energy. If the messages are sent continuously, the device has much more activity to perform so the power consumption is higher.

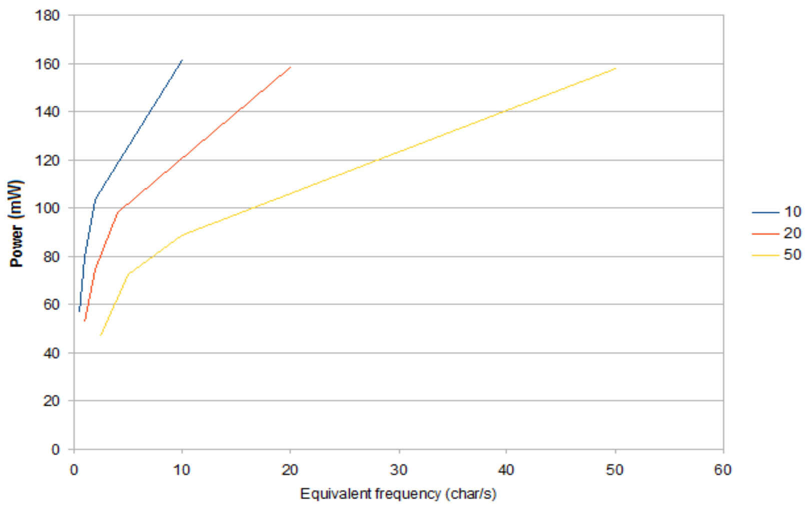


Figure 5.15: Power consumption depending on characters equivalent frequency when receiving messages.

Figure 5.15 shows the consumption depending on the equivalent frequency of characters sent per message. The conclusion is the same as in the time interval chart. We can see the same power consumption with 10, 20 or 50 characters per second regardless the lenght of the message.

5.10 Case 10. Instant Messages (Sending)

Finally, this case deals with different instant messages sent from the N810 to the second client with different lenghts and time intervals.

Chars	Interval (s)	Chars/s	Time (s)	Power (mW)	Energy (J)
5	5	1	71.40	169.55	108.97
5	10	0.5	71.90	137.71	87.90
5	20	0.25	78.30	113.83	84.82
10	5	2	88.10	188.94	131.08
10	10	1	78.10	154.35	102.58
10	20	0.5	80.30	131.14	92.03
20	10	2	73.20	146.26	105.48
20	20	1	82.40	141.55	103.80
20	30	0.67	85.10	139.08	100.41

Table 5.10: Measurement results in case 10.

The last use case, although it is analogue to the previous one, presents a big difference.

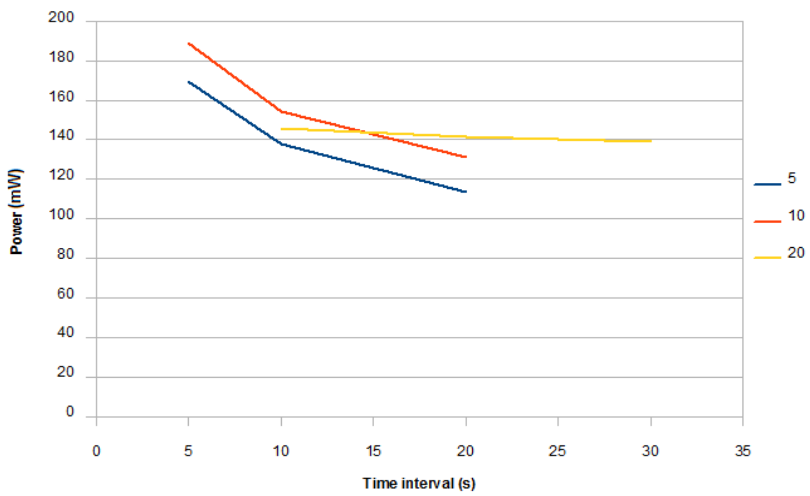


Figure 5.16: Power consumption depending on time interval between sent messages.

In case 9, the consumption is based on the messages received on the N810, but here, in addition to the power requirements of sending the messages, there is the power consumed while writing the characters of the message, since, as seen in the introduction of this chapter, every tap on the screen or key pressed on the N810 increases greatly the consumption.

On figure 5.16, the behaviour shown is that, like in case 9, the lower the time interval between messages, the higher the power consumption, but due to the message writing, the length of the message is also important. With messages of 5 and 10 characters, there is a difference of around 20 mW. With a longer message of 20 characters, the consumption varies much less than in the other cases, remaining more or less constant regardless the time interval. According to these results, a good recommendation for the N810 users is to write long messages, that causes less consumption than short ones, even if they are sent within small time intervals.

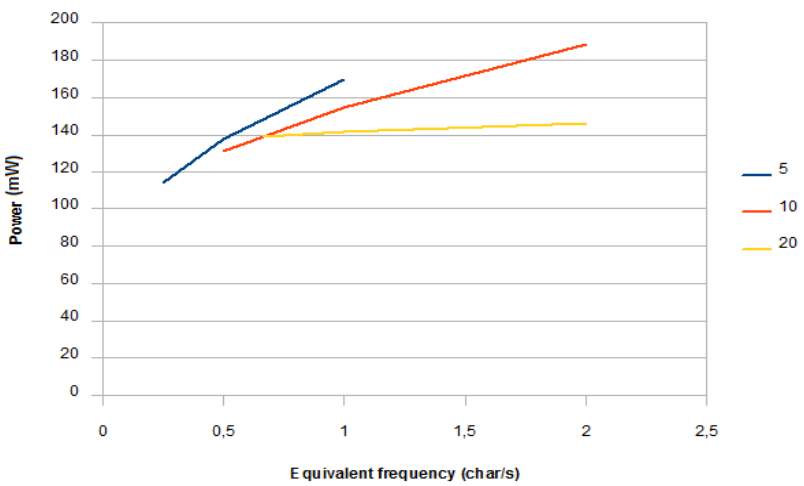


Figure 5.17: Power consumption depending on characters equivalent frequency when sending messages.

Figure 5.17 reveals the same results as the time interval one. With relatively small messages (5 and 10 characters), the total consumption is higher, although we can see that the consumption, even being higher with the 10-character message, increases slower than with the 5-character one. But again, the 20-character one presents the best behaviour, showing a very small variation and a lower average total consumption.

Chapter 6

CONCLUSIONS

This chapter summarizes the results obtained in the previous one, giving tips for a better use of the handheld device when it comes to save energy. Finally, also there are some suggestions to other topics about energy savings in mobile devices.

Many factors concerning the IM software and wireless networks have been analyzed in this study, but the most important one affecting the power consumption has been the time, which was not explicitly said to be analyzed. However, time itself is affected by other aspects which have been analyzed.

The introduction to chapter 6 showed a chart where the evolution of the power consumption during an IM session could be seen. In this chart, we could see that every time there was an event happening in the N810 (tapping on the screen, receiving network packets, ...) the consumption raised considerably, but reaching a maximum. Since this maximum had not had big differences regardless the activity of the device (at least with the use of CPU we caused, because there were no other user processes than the IM client activated on the N810), the main point of the power consumption is for how long the activity of the machine stays at maximum.

But of course, the time is affected by other factors, which in the end will affect the power consumption itself. In cases 1 and 2, the characteristics studied were the type and size of the files sent. The conclusions in this case were that the type is completely irrelevant since the files are sent inside MSG packets, which does not modify at all the internal structure of the file. However, the size affects greatly the power consumption. In the experiments, the transfer speed has been more or less constant during all the tests, so the maximum activity time of the machine depended on how big the file sent was. Obviously, the greater the size, the longer the time and so the power consumption.

After the size of the information sent, another important aspect that affects the consumption is the WLAN beacon interval. Small intervals force the wireless devices to have a great activity in order to catch and analyze the WLAN beacon frames. This causes the energy consumption to raise greatly, as could be seen in use cases 3 and 4. But having an extremely big beacon interval does not help the power consumption to be reduced. With a high interval, the interaction between the device and the wireless network is greatly reduced and, although the device can stay longer in power save mode, the transfer times increase a lot, especially when sending the information from the wireless device. This increase in the time causes the power consumption to raise even more than with a small beacon interval, since it keeps the device performing its maximum activity during longer time. The best beacon intervals observed were those between 20 and 100 ms.

Surprisingly, the WLAN transfer rate has not a great influence on the time nor the power consumption, although some differences were observed with the different settings in the experiments. Anyway, the power differences, in absolute values, were not as significant as with the beacon or the size, so we can conclude that the transfer rate is not a major issue affecting the power consumption.

The peer locations of the devices used in the IM have been analyzed, and, similarly to the transfer rate, the influence is not as big as with other factors. The results show a better behaviour when both devices were connected to the same access point in the same network, but this use of IM technology is very rare, due to the fact that the users are in most cases in different physical locations, what leaves this parameter completely out of control.

Finally, the last two use cases measured the behaviour of the power consumption with instant messages, not with file transfers, having different lengths and frequencies. The most interesting results were found here. First of all, the length of the messages affects very little when they have to be received in the N810, since any message is kept in only one single MSG packet, so the activity of the device only varies in reading and processing packets with differences of a few bytes. The behaviour changes if the messages are to be sent from the N810, since writing a message (either tapping on the screen keyboard or pressing the keyboard itself) causes a great activity in the N810 CPU, therefore, the power consumption increases with longer messages. In addition to the message length, the frequency of receiving or sending the messages also causes differences on the energy behaviour. A continuous usage, i.e., sending messages with very small time intervals between them keeps the device working at maximum. However, if there is a reasonable time between one message and another, the device is allowed, in one way or another, to "rest", so it is not performing a great activity during a continuous time and therefore, the behaviour is improved, reducing the energy consumption. An important issue concerning both aspects (length and frequency) is that longer messages cause the energy to vary much less than shorter ones, regardless the time interval.

Possible future work about energy saving with IM in mobile devices could deal with, now that this study presents how the different factors affect the energy, designing software that tries to avoid the conditions that have been proved to be worse, like extremely big files, high beacon intervals and continuous usage of the device. The problem with big files could be solved by compressing them, or avoiding the transfers if the speed is very low. The beacon interval is something out of reach for the IM software, since it concerns the administration of the whole network. And also the user behaviour of the handheld device is difficult to control for the software, since, even if the program does not allow to send very small messages with a high frequency, the user might be tapping the screen or using the keyboard anyway, what would cause a higher consumption even if the IM software is not being used.

Bibliography

[1] Jaganath Achari. Nokia N810 vs the iPod touch vs the Asus EeePC, 2007. http://www.jaganath.net/content/linux/nokia-n810-vs-the-ipod-touch-vs-the-asus-eeepc/75.

[2] Dovgan Alexander. Development and studying of power-saving transformer for a management of the automated electric drive for mining hosting plant. Donetsk National Technical University, 2007.

[3] Ola Andersson, Niclas Larsson. RSS - The Future of Internal Communication?. Växjö Universitet, 2005.

[4] Naeem Zafar Azeemi. An Energy Aware Framework for Mobile Computing. Technischen Universität Wien, 2007.

[5] Jakop Berg, Martin Kurtsson. A position aware mobile instant messaging client. Umeå University, 2008.

[6] Md. Rafiqul Hasan Chowdury. UPnP Connectivity between home networks. Helsinki University of Technology, 2007.

[7] M. Day, J. Rosenberg, H. Sugano. A Model for Presence and Instant Messaging, 2000. http://www.ietf.org/rfc/rfc2778.txt.

[8] Yunsi Fei, Lin Zhong, Niraj K. Jha. An energy-aware framework for dynamic software management in mobile computing systems. ACM, 2008.

[9] Fluke Corporation. FlukeView Forms Documenting Software. Users Manual, 2003.

[10] Jim Geier. 802.11 Beacons Revealed, 2002. http://www.wi-fiplanet.com/tutorials/article.php/1492071.

[11] Jim Geier. Ending the 802.11 Network Card Power Drain, 2002. http://www.wi-fiplanet.com/tutorials/article.php/1015781.

[12] IEEE. IEEE 802.11g-2003: Further Higher Data Rate Extension in the 2.4 GHz Band, 2003.

[13] Michael Kassner. Wi-Fi Power Management: Road warriors beware, 2008. http://blogs.techrepublic.com.com/networking/?p=533.

[14] Daniel Kelley. The Hybrid Structure of Instant Messaging. Middlebury College, 2008.

[15] Hyun Suk Kim. Energy-aware hardware and software optimizations for embedded systems. The Pennsylvania State University, 2003.

[16] Hemant Kumar. Energy Aware Security Services for Sensor Networks. University of Massachusetts, 2005.

[17] Bill McFarland. How to use optional wireless power-save protocols to dramatically reduce power consumption, 2008. http://www.wirelessnetdesignline.com/showArticle.jhtml?articleID=208400363.

[18] Mike Mintz. MSN Protocol, 2002-2004. http://www.hypothetic.org/docs/msn/.

[19] Nate Mook. MSN Messenger Most Used IM Client, 2006.

[20] Nokia Corporation. Nokia N810 Reference Manual, 2007.

[21] Peter Salin. Mobile Instant Messaging Systems - A Comparative Study and Implementation. Helsinki University of Technology, 2004.

[22] Hiren Kumar Deva Sarma, Avijit Kar, Rajib Mall. Energy Efficient Communication Protocol for a Mobile Wireless Sensor Network System. International Journal of Computer Science and Network Security, 2009.

[23] Tanwir Sheikh. Modeling of Power Consumption and Fault Tolerance for Electronic Textiles. Faculty of the Virginia Polytechnic Institute and State University, 2003.

[24] Scott Sidel. Wireshark: Taking a bite out of packet analysis, 2007. http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci1246794,00.html.

[25] Fredrik Stålnacke. Implementation of an Instant Messaging Client using the OMA IMPS Protocol. Umeå University, 2003.

[26] Juha Väisänen. Configuration Management for Performance Reporting in Telecommunication Networks. Helsinki University of Technology, 2005.

[27] Kia Wood. Instant Messaging Usage and Academic and Social Integration. Virginia Polytechnic Institute and State University, 2007.

[28] Shushan Zhao. Connectivity Management in ad hoc Networks and its Implementation in WIDENS Network. Helsinki University of Technology, 2005.

Appendix A

MSNP COMMANDS DESCRIPTION

This chapter describes each MSNP command shown in chapter 3, explaining their objective more deeply, as well as other aspects such as the parameters they use.

A.1 VER

The VER command is used to specify which version (or versions) of the MSNP is supported. The purpose of the command is, on the client side, notify the server which versions can support, and on the server side, choose one of the versions the client sends, and establish it as the one used in the communication. It could happen that the server does not recognize any of the versions that the client sends, in which case the connection will be closed immediatdly.

A.2 CVR

CVR sends from the client to the server some information about the client version and the operating system in use. This command has a TrID and eight parameters. They are shown below.

1. Locale ID (language code), in hexadecimal format.

2. Operating system type

3. Operating system version
4. Architecture of the computer in use
5. Client name
6. Client version
7. The string "MSMSGS", reasons unknown
8. Account name

The server's response is another CVR command, but this time with only five parameters.

1. Recommended version to use
2. Identical to the first parameter
3. Minimum safe version
4. A URL where it is possible to download the recommended version
5. A URL with additional information about the client

A.3 USR

This command is used to notify the server the account name we want to use to log in. It must be sent after CVR. The parameters required are the following.

1. Authentication system, always "TWN"
2. Letter 'I' (standing for *initiating* authentication)
3. Account name to log in (same as in CVR)

The most common response from the server will be an XFR command (but with the same TrID), or an error if the server can not accept the connection.

A.4 XFR

This command is always a response to a CVR. Its purpose is to redirect the client into a different server. The reason for this is the separation between "dispatch servers" and "notification servers". First ones are used mainly for authentication which will later redirect the clients into one of the second, which are the ones that allow the exchange of messages.

XFR has the same TrID as the previous CVR and four parameters.

1. The string "NS", to let the client know that will be transferred to a notification server

2. IP address and port of the notification server

3. Always number 0

4. IP address and port of the current server

A.5 SYN

The SYN command is used to synchronise the contacts list. It only has a TrID and one parameter (the version number of the list stored in the client, that may be 0 if the client does not store personal information, which Pidgin does).

Depending on the version number the client sent, the server response may be one or another. If the version number the server has matches the client's, the response will just contain the TrID and the same number, but if it does not, the server will get ready to send all contacts information (as well as if the number was 0).

The contacts list will be synchronised using several additional commands.

A.5.1 GTC

Determines the action to perform when someone adds us to his/her contacts list. The default value is 'A', which means that this new contact will be added to our own contacts list. The only different value is 'N', that will block the new contact, but also will send a notification to the client.

A.5.2 PRP

It is used to synchronise stored phone numbers. It has not been used in the tests of this study.

A.5.3 LSG

Synchronises the groups defined for our account. The server sends a LSG per group with the following parameters.

1. Number of the group
2. Name of the group
3. Another number which meaning is unknown, but it is always 0.

A.5.4 LST

Synchronises the contacts one by one. The server sends a LST per contact with the following parameters.

1. Contact's account name
2. Nickname
3. Lists this contact is in (deals with notifications about status changes)
4. Number of the group the contact belongs to

A.5.5 BLP

Controls the blocked contacts. The BLP command sent to the server contains a TrID and a value which may be "AL" or "BL". If "AL", the contacts who are not in our list will be allowed to know our current status and chat with us, which will not be allowed if the value is "BL".

A.6 CHG

Sets the user's presence status. It has a TrID and two parameters.

1. A three letter code indicating the status
2. The client ID number

If it is successful, the client will reply with exactly the same command and parameters.

Some codes for the status are NLN (online), AWY (away) or HDN (hidden).

A.7 NLN

Unlike the NLN code used in the CHG command, NLN is also a different command, used to notify a status change of one of our contacts. In this case, an NLN command will always be sent from the server to the client, since it is the server the one who knows about the contacts' status changes. NLN has no TrID, but it has four parameters.

1. Status code (same as in CHG)
2. Account of the contact that changes the status
3. Nickname
4. Contact's ID number

Since NLN is both command and status code, it is perfectly possible to receive a command like the following.

NLN NLN friend@domain.net TrendyNickname 1234

This command notifies us that our friend "TrendyNickname", whose account is "friend@domain.net" is now online.

A.8 FLN

As well as NLN (the command, not the status code), it is sent by the server to the client. In this case, it notifies that a contact has closed the connection (or has changed the status to hidden, but this is not notified with an NLN command). It has not TrID either, but unlike NLN, only has one parameter, which is the account address of the contact.

A.9 Pings / Keep-alive signals

PNG (standing for "ping") are commands sent from the client to the server to let this one know that our status has not changed (i.e. we are still connected). They are usually known as keep-alive signals in other areas. Pings are sent with no TrID or parameters.

QNG is the server's response to a client's PNG. This QNG response does not contain a TrID either, but there is a parameter which is always a number with unknown meaning.

In addition to PNG and QNG there is a pair of commands that also deals with keep-alive signals, but in this case, it is the server who sends them at first. In order to make sure a client is still connected, the server sends a CHL command.

CHL has two parameters. First one is always number 0 and the second one is a 20-digit number.

If the client wants to keep connected, it is mandatory to respond the CHL with a QRY before 50 seconds pass since the CHL was sent, otherwise the server will consider the client disconnected.

QRY has a TrID and three parameters:

1. The client ID number
2. A special code that is the MD5 digest of the 20-digit number received with the CHL command
3. The length of the previous code, which is always 32 (bytes)

Once the server receives the correct QRY, it responds with another QRY and the same TrID, but no parameters this time.

A.10 OUT

This command closes the connection with the server. It has no TrID or parameters and once the server receives it from the client, it will immediatly close the connection without responding anything.

A.11 Switchboard

There are three MSNP commands involved in the creation of a switchboard session. A switchboard session can include several different users, but the behaviour is the same regardless the number.

The first command to explain is **RNG**. It is sent to those clients who will be involved in the switchboard session. The parameters are the following (there is no TrID).

1. Switchboard session ID
2. IP address and port of the server
3. Type of authentication, always "CKI"
4. Authentication string
5. Account name of the user who started the session

After the user receives the RNG, it must respond with a **ANS** command, which will let the server know that we are ready to start the switchboard session. It has a TrID and the following parameters.

1. Our own account name
2. Authentication string (must be the same that was received in the RNG)
3. Switchboard session ID (the same as in RNG as well)

After this is done, the server will confirm the ANS with one or more **IRO** commands. The number of IROs depends on how many participants are joining the switchboard session, exactly one IRO per user. It has the same TrID as the ANS, and the parameters are the following.

1. Number of the current IRO command. It starts with 1 and increases with each IRO received.

2. Total number of IROs that will be received (i.e. total number of participants in the switchboard session, excluding ourselves)

3. Participant's account name

4. Participant's nickname

After the IRO, the server will respond with another ANS command and still the same TrID, but this time the only parameter will be the string "OK".

Once this is done, the message exchange can start, using MSG commands. When one of the user wants to finish the switchboard session, an OUT command is sent.

The following diagram shows the command exchange between the server and two clients when establishing a switchboard session.

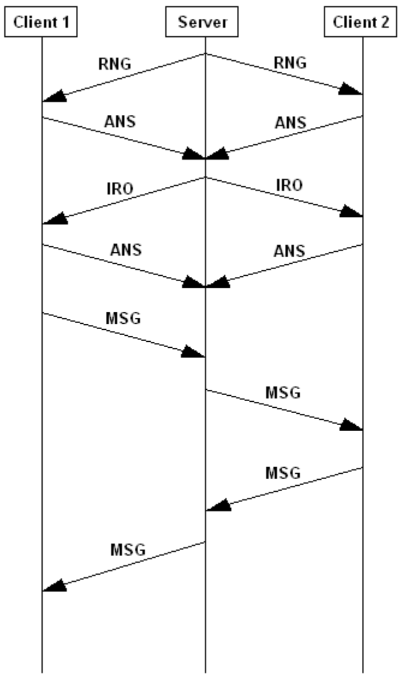


Figure A.1: Command sequence in a switchboard session.

A.11.1 Sending messages

The way to send text messages from one client to another is the MSG command. MSG is a payload command, which has a different treatment, as explained early in this chapter. As a payload command, it has the length of the payload specified as a parameter, in order to know where is located the end of this payload, which in fact contains the written message. It has a TrID and the following parameters.

1. Type of acknowledgement
2. Payload length (in bytes)
3. Payload (the written text)

About the type of acknowledgement. This parameter can be set to 'U', 'N' or 'A'. The value determines the behaviour of the other client when it receives the message, which can be:

- If 'U', the receiver will send no acknowledgment
- If 'N', the receiver will send acknowledgment only if the message was not received properly
- If 'A', the receiver will always send acknowledgment.

Acknowledgement responses can be done either with ACK, which means the message was delivered correctly (only if the type is set to 'A') or with NAK, which means the message was not delivered the way it should (can happen with 'U' or 'A').

In the experiments, sometimes a fourth acknowledgment type, 'D', has been observed too, which behaviour seems to be exactly as if it was 'A'.

A.11.2 Receiving messages

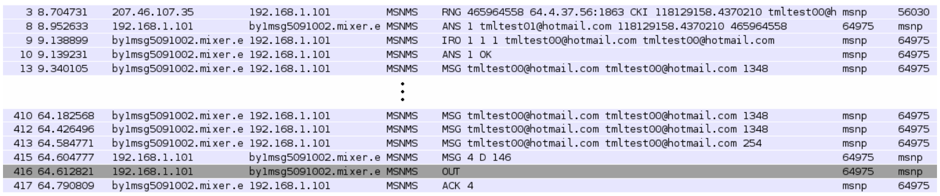
In order to receive messages, the MSNP command used must be MSG as well, but it is slightly different than the MSG command used while sending messages. It still has a TrID, but the parameter structure changes like this:

1. Sender's account name

2. Sender's nickname
3. Payload length (in bytes)
4. Payload (the message text)

A.11.3 File Transfer Example

Here we can see a file transfer example. The image shows the packets captured with Wireshark when receiving a file in the N810.



3	8.704731	207.46.107.35	192.168.1.101	MSNMS	RNG 465964558 64.4.37.56:1863 CKI 118129158.4370210 tmltest00@h	msnp	56030
8	8.952633	192.168.1.101	by1msg5091002.mixer.e	MSNMS	ANS 1 tmltest01@hotmail.com 118129158.4370210 465964558	64975	msnp
9	9.138899	by1msg5091002.mixer.e	192.168.1.101	MSNMS	IRO 1 1 1 tmltest00@hotmail.com tmltest00@hotmail.com	msnp	64975
10	9.139231	by1msg5091002.mixer.e	192.168.1.101	MSNMS	ANS 1 OK	msnp	64975
13	9.340105	by1msg5091002.mixer.e	192.168.1.101	MSNMS	MSG tmltest00@hotmail.com tmltest00@hotmail.com 1348	msnp	64975
				⋮			
410	64.182568	by1msg5091002.mixer.e	192.168.1.101	MSNMS	MSG tmltest00@hotmail.com tmltest00@hotmail.com 1348	msnp	64975
412	64.426496	by1msg5091002.mixer.e	192.168.1.101	MSNMS	MSG tmltest00@hotmail.com tmltest00@hotmail.com 1348	msnp	64975
413	64.584771	by1msg5091002.mixer.e	192.168.1.101	MSNMS	MSG tmltest00@hotmail.com tmltest00@hotmail.com 254	msnp	64975
415	64.604777	192.168.1.101	by1msg5091002.mixer.e	MSNMS	MSG 4 D 146	64975	msnp
416	64.612821	192.168.1.101	by1msg5091002.mixer.e	MSNMS	OUT	64975	msnp
417	64.790809	by1msg5091002.mixer.e	192.168.1.101	MSNMS	ACK 4	msnp	64975

Figure A.2: Packets captured when receiving an example file.

As shown in the figure, the user receives a RNG command in order to start a switchboard session (with the necessary ANS and IRO afterwards). Below are shown all the MSG packets that contain the file and in the end, the OUT command is sent to finish the transfer, with acknowledgement from the sender.