Bluetooth Broadcasting

Or

Designing a scalable system to distribute context aware information to mobile groups via Bluetooth

A feasibility study

Supervisors Prof. Dr. M. Aiello
R*u*G

Ir. S. Achterop
R*u*G

Author Remko de Jong
S1277081

Date 17 July 2009

Version 1.0

Abstract

As a short-range wireless protocol, Bluetooth was originally intended as a cable replacement. Nowadays, Bluetooth has outgrown its original purpose and is now also used for a variety of other applications. A growing domain is its use for proximity marketing, i.e. the localized wireless distribution of advertising content associated with a particular place. Content, varying from simple text messages to MP3, video, business cards, or pictures are broadcasted to passing clients. Companies trying to ride this hype have popped up like mushrooms over the last three years, but almost no open research to the performance and scalability of these systems has been done. How do these systems cope with large groups of mobile users, possibly unaware of the fact someone is trying to communicate with them? In this thesis we investigate what the hardware and software requirements are to set up a scalable message distribution network to distribute location-based information about exams, lectures and daily news within the faculty of Computing Science of the University of Groningen to large mobile groups equipped with Bluetooth-enabled devices. We take a look at the possibilities to handle more than seven active simultaneous Bluetooth connections, what kind of strategy to use and the hard- and software limitations that come with the various implementations of the Bluetooth protocol stack.

Keywords: *Bluetooth, Broadcasting, Scalability, Proximity marketing, Context-awareness, Distributing to mobile groups, Pervasive computing, Seven active connections limitation*

Table of Contents

Figures and Tables

Figures

Tables

Preface

Times are changing. If computers were a luxury item some decades ago, only available for the happy few who had enough money and a firm desk to put these colossal beasts on, nowadays they come in such large numbers and such small sizes, that most of the times we do not even realize that they have infiltrated into almost any part of our life. However, this pervasiveness has opened the door to some very interesting research and alternative uses of common technologies in an unexpected way.

One of these examples is the use of the short-range wireless protocol that goes by the name of Bluetooth. In this thesis we use the protocol to broadcast content to mobile groups equipped with Bluetooth-enabled devices and investigate the problems that came up along the way. The purpose of this research is to explore the possibilities and limitations of a distribution system that delivers free information about exams, lectures and daily news to students based on their location.

The electronic learning environment of the University of Groningen is called Nestor. Teachers use Nestor to give information about courses, to take electronic examinations, and to exchange documents with their own department. Students use Nestor to read announcements, to cooperate with group members, and to hand in assignments. The Nestor digital blackboard system is the main candidate for providing the content that is to be distributed among the students. Of course, this does not mean that in the future the system cannot be extended to distribute other content. Downloading PowerPoint presentations to your mobile phone during a lecture is just one of the many applications the system can be used for.

To get a better understanding of the context we are working in, we first explore the current pervasiveness of computing, and get ourselves acquainted with the Bluetooth protocol (Chapter 1). Next we do an in-depth study of current state-of-the-art for this domain (Chapter 2) after which we can focus on the gap that lies between previous research and the research conducted in this thesis (Chapter 3). Based on the requirements for such a scalable distribution system, we designed and implemented a system to be able to put all our questions to the test. The architecture and implementation of this system are discussed in Chapter 4 and Chapter 5. A description of the different experiments can be found in Chapter 6 and the results and the discussion of the results are presented in Chapter 7 and 8. Finally, suggestions for further research are done in Chapter 9.

Acknowledgements

I would like to thank the following people, since this thesis would not have gotten the form it is in now if it where not for their contributions and support. First of all I would like to thank my supervisor, Marco Aiello, for his continuing support and advice and for having enough interest in the subject to be willing to pursue publication. Besides him I would like to thank Joël de Nes for his contributions in the form of the implementation of the software and for the fact he was willing to implement the changes I never stopped demanding. I also thank Sietse Achterop for willing to be my second supervisor and his contributions. Then my gratitude goes out to Simon Dalmolen, who worked on the predecessor of this project and helped me with advice and testing. Finally I would like to thank Tim de Jong and Frank Noteboom for their extensive review of this thesis, enabling me to take it up another notch.

Groningen, July 2009,
Remko de Jong

Part I
Introduction

1 Pervasive computing

Pervasive computing, ubiquitous computing, ambient intelligence, or *everyware,* is a relatively new (1988) concept describing the integration of ICT into people's lives and environments. It has also been described as the third wave (cf. Table 1-1) of computing technologies[1].

Wave	Description
First wave	Mainframe computing era. One computer is shared by many users by means of workstations.
Second wave	Personal computing era. One computer is used by one user. Interaction is conscious and usually bound to one computer and peripherals on the desktop.
Third wave	Pervasive computing era. One person, many computers. The growing availability of increasingly smaller microprocessors with communications facilities makes it possible for computers to be embedded in the background, allowing unconscious user interaction with them to aid everyday tasks.

So the main property of pervasive computing is that it disappears into the background, silently waiting there until we need it. An example of pervasive computing is altering lighting or music in reaction to the location and activity of a user. As you leave the room the lights are dimmed and the lights in the room you are entering are turned on without you having to do anything conscious.

The application domain for pervasive computing is a vast field. The most common are healthcare, home care, transport and environmental monitoring. In healthcare patients can be monitored using wearable sensors to detect irregularities in behavior and alert the medical staff. Population ageing offers a great perspective for home care. The number of people aged 65 years and older in Western countries is increasing and these people will increasingly require care from a diminishing working population. Sensors can be embedded in homes to monitor irregularities in temperature, on the body to monitor heart rate and visual displays or voice messages could be used to remind people to take their medications. Environmental monitoring requires computers to be lightweight, power effective, and expendable, since they usually have to withstand harsh environmental conditions such as heat, cold, or rain. Monitoring near volcanoes is one example. Usually environmental monitoring is done over a long period of time. Finally, parking sensors and lights that adjust are two examples of computers already integrated into modern cars. We can even take this one step further by adding an 'intelligent' component to enable vehicles to exchange information while on the move. Working together with devices embedded into transport infrastructure such as signs along the road, drivers can be alerted of traffic congestions, accident hotspots and road obstructions[1].

1.1 Spontaneous networking

A term closely related to pervasive computing is that of *spontaneous networking*. Usually, more than one embedded device is needed to perform a certain task. In order to perform this task the devices need to communicate with each other. To communicate the devices 'hook up' with each other and form a network. Because pervasive computing is everywhere you cannot be sure which devices are at hand, which means you will have to deal with your environment and set up an ad hoc or spontaneous network.

Because of the mobility of the user (and of the devices of course) it is seemingly unknown at any given moment which devices in the surroundings can be used to perform a certain task. So there must be a means to discover which devices are in the neighborhood and ready to be used. This is called device discovery.

Some devices provide an interface where the services the device offers can be retrieved. Once the various devices have been discovered it is essential to know what tasks or services they are built to perform. This process is called service discovery. For example, a stereo set can be used to play music, but not to turn down the lights. Every device is built with a purpose and service discovery is about finding out what their purpose is, so we can use it in the right way.

Devices need to know where they are in the network in order to provide the right information in the right situation. This is called context awareness. It means that it is important for some services to know where you are in order to give you relevant feedback. Imagine a service that provides tourists with background information about the city they are visiting. If you are visiting Amsterdam, it is of little use to you to know where the best pubs in Dublin are situated.

1.2 Bluetooth

Bluetooth wireless technology (IEEE 802.15.1) is a short-range communications technology originally intended to replace the cables connecting portable and/or fixed devices while maintaining high levels of security. The key features of Bluetooth technology are threefold: robustness, low power, and low cost. Bluetooth has been designed in a uniform way. This way it enables a wide range of devices to connect and communicate with each other by using the Bluetooth wireless communication protocol[3].

The Bluetooth technology has achieved global acceptance in such a way that any Bluetooth-enabled electronic device, almost everywhere in the world, is able to connect to other Bluetooth-enabled devices in its proximity. Bluetooth-enabled electronic devices connect and communicate wirelessly through short-range, ad hoc networks known as piconets. Each device can simultaneously communicate with up to seven other devices within a single piconet. Each device can also belong to several piconets simultaneously. Piconets are established dynamically and automatically as Bluetooth-enabled devices enter and leave radio proximity[3].

One of the main strengths of the Bluetooth wireless technology is the ability to handle data and voice transmissions simultaneously. This enables users to use a hands-free headset for voice calls, printing, fax capabilities, synchronizing PDA's, laptops, and mobile phone applications to name a few[3].

1.3 Scalability of Bluetooth broadcasting

An important aspect of this thesis is about the scalability of Bluetooth broadcasting. Since scalability can sometimes be a rather vague concept, we give a short explanation of the term. An important aspect of software products is how they are able to cope with growth. For example, how does the system handle an increase in users or data traffic? This property of a software system is usually referred to as scalability. A more detailed specification of the concept is given by André Bondi[4], who defines it as follows: 'Scalability is a desirable attribute of a network, system, or process. The concept connotes the ability of a system to accommodate an increasing number of elements or objects, to process growing volumes of work gracefully, and/or to be susceptible to enlargement.' Whenever a system meets these requirements we can say that the system scales. In this thesis scalability comes down to the question if the system is capable of dealing with large groups of users equipped with Bluetooth-enabled devices capable of receiving simple text messages.

1.4 Proximity marketing

One of the latest trends in advertising is called proximity marketing. Proximity marketing is the localized wireless distribution of advertising content associated with a particular place. One way of transmitting the messages is broadcasting them to nearby devices via the Bluetooth protocol. This broadcast can vary from simple text messages to multimedia content such as video, business cards or applications. Below are a couple of examples to clarify this further.

Example 1.
Imagine you are walking through a mall and you pass a proximity broadcast station. The key is to have your phone on and in 'discoverable' mode. Discoverable mode means that your phone can be found by inquiring devices. It is visible so to speak. This will allow all possible ads in the area to 'hit' your phone – asking if you want to receive free content from the provider. For example, say you are shopping for gym equipment at a specialty shop and they have a proximity marketing station set up. If your phone is in discoverable mode, you will receive a message asking if you want to receive free content from 'Company X'. You can think of it as a virtual billboard or flyer advertisement, but now the distribution goes via Bluetooth[5].

Example 2.
During the summer of 2005, the British rock band Coldplay used BlueCasting[18] (a Bluetooth proximity marketing system) to promote its newly released album X&Y. During a two-week period, approximately 20,000 people downloaded pre-release video clips, never-before seen interviews, audio samples and exclusive images directly from posters in the main rail terminals of London by using their mobile phones or other Bluetooth-enabled devices[6].

At the moment, the number of companies trying to ride this hype is overwhelming. Advatex, Alterwave, Assertivemedia, Blipsystem, Bloozone, Bloozy, Bluead, Blueblitz, Bluebot, BlueCasting, Bluecell, BlueGiga, Bluehotspot, Bluepulse, Bluetooth-Advertising, Bluetotem, Breeze-tech, CmoGlobal, Futurlink, Goyya, Halfbakery, Hypertag, Jellingspot, Kameleon, Midray, Norkatech, Panther Bluetooth, Proxi-ma, Proximitymedia, RTX, Smart and wireless, WCIT and Zonablu are just a few in a field of many. Some of these companies will be discussed in greater detail in paragraph 2.3.1 which deals with commercial projects in the Bluetooth broadcasting sector.

For now we focus on discussing the different forms of proximity marketing that are possible today. We can make a division between two types of companies: companies that sell equipment and / or software to do a broadcast from one or more central locations and companies that are recruiting customers to become Bluetooth broadcasters themselves by paying them on, for instance, a pay-per-ad basis.

1.4.1 Passive broadcasting

The first type of business deals with broadcasting from a central location, which we will call passive broadcasting. Most of these companies sell both the hardware and software to enable this. For example, BlueCasting by Filter WorldWide, one of the major players in the market which made the news in August 2005 when they distributed merchandise for the British pop band Coldplay, offers a product family divided into four types of systems. They offer solutions for small retail shops, one-off events such as music festivals, and even larger areas such as airports and train stations. The latest descendant in the family is a system that provides an interactive touchscreen allowing users to interact directly with the system. BlueCasting is an example of a product that comes with both hardware (one or more BlueCast Servers) and software (BlueCast Campaign Management System) which is used to provide remote setup, maintenance and reporting.

Besides this type of companies, i.e. the ones that are selling the total package, other companies have dedicated themselves to providing just the hardware. An example is BlueGiga. According to their website their BlueGiga Access Servers are used by more than 350 Bluetooth Marketing companies in more than 65 countries[8]. They sell two lines of products: Bluetooth Modules and Bluetooth Access Servers. The modules are described as 'completely integrated, certified, high-performance Radio Frequency products including all needed Bluetooth profiles'. Access Servers are sold in the form of Access Points (up to 7 connections) and Access Servers (up to 21 connections). Besides this they also sell the BlueGiga Solution Manager (BSM). This is a web-based remote management and monitoring platform for BlueGiga Access Servers that can be used to simultaneously upgrade, monitor and configure a large number of BlueGiga Access Servers, instead of configuring each device one-by-one.

1.4.2 Active broadcasting

A different use of marketing via Bluetooth broadcasting is put into practice by ZoomBroadcast[7]. This company offers consumers to make money by becoming either a passive or an active broadcaster. The main idea is that

1) a consumer buys one or more Bluetooth Broadcasters,
2) then either finds one or more places in their city to locate them or puts one of the Broadcasters in his or her bag, purse, backpack, or coat and goes to a football game or an outdoor concert, and
3) finally collects royalty checks based on how many ads are delivered through the Broadcaster(s) to the mobile phones of other people.

1.4.3 Statistics

To show why proximity marketing is such an interesting business, some figures[9] are presented in this chapter. According to an independent survey of 50 brand names performed by Airwide Solutions the following statistics apply:

- Over 200 million Americans carry mobile phones. This is more than half of the country's population.
- Cell phones are used by over 3,1 billion people globally.
- 89% of major brands plan to market via mobile phones by 2008.
- 40% of major brands have deployed text messaging (SMS) campaigns.
- 18% of major brands have deployed multimedia messaging (MMS) campaigns.

Other research in the field of mobile marketing done by M:Metrics[1] gives us more insight in the value of this market.

- The global mobile advertising market will be valued at over $16 billion by 2011.
- In August 2007, nearly 40 million US consumers received SMS advertisements, and 12 percent responded to them.

Another interesting fact:

[1] Source: M:Metrics, Common Short Codes: Cracking the Mobile Marketing Code

- A survey amongst 2,400 mothers reveals that the single most important tech gadget in their lives is the cell phone (23%), followed by the Internet (21%) and the digital camera (19%).

Of course this sounds promising, but for this thesis it is also interesting to know some statistics about Bluetooth for all these mobile phone users. BlueMediaServer, a company that is into active broadcasting provides some statistics about broadcasting via Bluetooth from a central location. They claim the following[10]:

90% of all users have Bluetooth turned off. From the 10% that remains, when asked if they will receive a file, 75% will say no, and of the 25% that says yes, 50% of the times the transmission is dropped because the mobile phone cannot communicate well from a large distance. So if you want to broadcast messages via Bluetooth from a central location, you should expect a hit rate of less than 1.25% of all possible clients.

Of course these are just statistics and the desire of the user to receive the content also plays a vital role. Remember the Cold Play campaign mentioned above for instance. When users initiate the transmission, they will probably not wander about but stay close to the antenna in order to maximize their chance of successfully receiving the content they long for. Especially with regard to the future – new phones have Bluetooth implemented as a standard – where technology keeps improving as well as (probably) the number of people with mobile phones, this promises to be a very interesting market.

2 State-of-the-art

Pervasive computing is the object of a lot of research. In this chapter we present an overview of the largest and most relevant projects in this domain as well as a discussion of more specific aspects of pervasive computing that are important for this thesis. These aspects include short-range wireless communication protocols and current applications that are based on Bluetooth broadcasting.

2.1 Relevant projects

In the following two paragraphs we discuss two large projects in the domain of pervasive computing. These projects are called Oxygen[11], a project by MIT, and Aura[12], a project by Carnegie Mellon University.

2.1.1 Project Oxygen

Oxygen is a project in development at MIT. Oxygen embraces the vision that 'in the future computation will be human-centered and freely available everywhere, like batteries and power sockets, or oxygen in the air we breathe.' Instead of learning new computer jargon, people will communicate naturally with their electronic counterparts, using speech and gestures to describe their intent and leave it up to the computer to carry out their will.

Project Oxygen faces a number of challenges. For a start it must be everywhere (*pervasive*) and incorporated into our world (*embedded*). Given our freedom to move where we please it must allow users and computations to move around as freely as humans do (*nomadic*) and thus be flexible and spontaneous (*adaptable*) to the ever changing environment of the user. It must free itself from constraints imposed by bounded hardware resources, addressing system constraints imposed by user demands and available power or communication bandwidth (*powerful, yet efficient*), and must be able to understand the intent of the people when addressing matters, for example 'the nearest printer' as opposed to by address (*intentional*). Finally it must be available at all times and never shutdown or reboot (*eternal*).

To meet all of these requirements Oxygen follows the following approach. It enables pervasive, human-centered computing through a combination of specific user and system technologies. By addressing human needs directly through the use of user technologies such as speech and vision technologies it enables us to communicate with Oxygen as if we are interacting with another person, saving much time and effort. Automation, individualized knowledge access, and collaboration technologies help the user to perform a wide variety of tasks. User information can be gathered via handheld devices, monitoring by cameras, speech recognition, and embedded devices with sensors after which more common communication lines can be used to distribute the information even further (Figure 2-1).

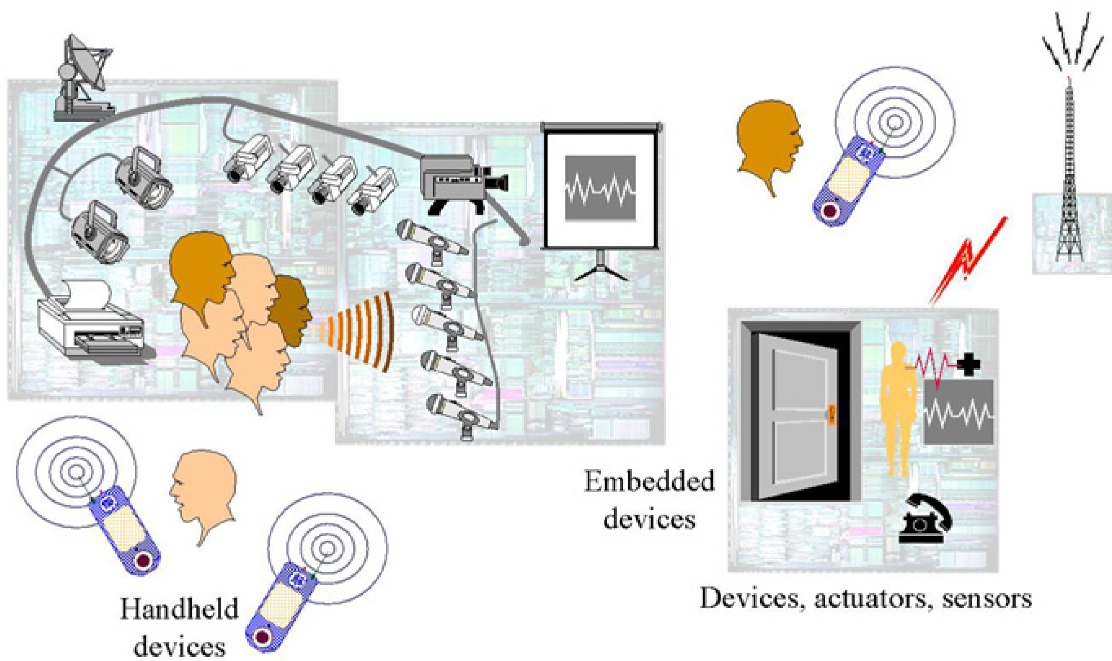


Figure 2-1: Gathering of user information in project Oxygen

Technological Overview

The technologies used in Oxygen are spread across different areas such as devices, networking, software, speech recognition, visual tracking, and user technologies.

The devices in Oxygen are comparable to batteries and power outlets in a way that they provide the power for computation, communication, and perceptions just as the aforementioned supply power for electrical appliances. Collections of embedded devices, called E21s, are used to create 'intelligent spaces' inside buildings, homes, vehicles, etcetera. Handheld devices (H21s) accept speech and visual input and can be used as cellular phones, beepers, radios, televisions, geographical positioning systems, cameras, or personal digital assistants. By offloading communication and computation to nearby E21s they are able to be more power efficient.

Network technologies (N21s) make up the glue that holds these devices together in order for them to form collaborative regions. The technologies used must be able to deal with dynamically changing configurations of devices.

The software environment facilitates change. The software architecture of Oxygen relies on controlling and planning abstractions that provide mechanisms for change that is caused by, for example, explicit user requests, current operating conditions or available upgrades.

In Oxygen interaction through speech and vision is preferred to conventional methods such as keyboards and mouses. Recognizing facial expressions and lip movement are examples of perceptual technologies used to augment speech understanding. An example of vision technology is a real-time object tracker that is able to track rotation and translation. Combining this with a face detector, the system is able to accurately track head positions, effectively enabling applications to perceive where people are looking.

User technologies are divided into automation technologies enabling the automation of repetitive information and control tasks, collaboration technologies to accommodate the formation of spontaneous collaborative regions and knowledge access technologies to improve access to all sorts of information.

2.1.2 Project Aura

The tagline for the Aura project is ‘Distraction-free Ubiquitous Computing’. Aura has been in development at Carnegie Mellon University since the year 2000 and its goal is ‘to provide each user with an invisible halo of computing and information services that persists regardless of location’. The developers of Aura claim that nowadays the most important resource in a computer system is *user attention* rather than conventional metrics such as memory, processing power or disk space.

As we look around us, we are surrounded by a wealth of computing, informational, and communication resources, all supposed to allow us to work more effectively. However, because of the variety and the sheer number of resources available, the current- day challenge has shifted to finding ways to harness this power without overburdening the user with the management of the underlying technology and infrastructure. Especially the factor that resources are varying in availability over time is important. Users may move to a different location, rendering some resources inaccessible whereas other resources are becoming available at the same time.

Project Aura takes a new approach to overcome this challenge. The solution is based on the concept of a personal 'Aura'. The idea behind the personal Aura is that it acts as a proxy for the person it represents.

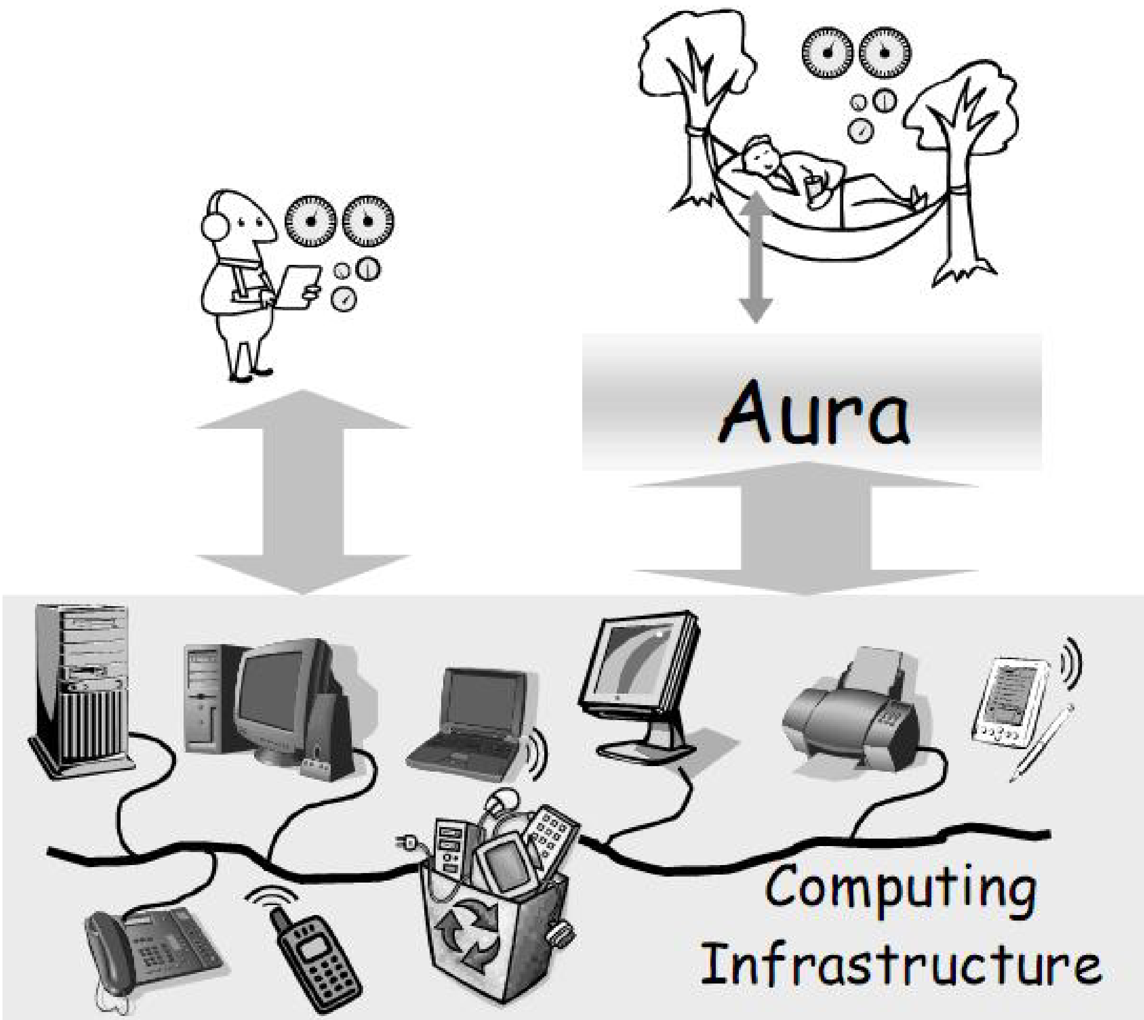


Figure 2-2: The Aura Approach

When a person enters a new location, her Aura will find the necessary resources that will help the person to complete her tasks. The Aura is meant to function as a layer between the person and the computing infrastructure and to catch the changes in the environment, this being the variability of computing environments as well as the instability of resources. Besides this, the Aura can also try to predict the future needs of its user. Figure 2-2 illustrates this idea. The thick arrow symbolizes the distracting interactions between a user and the pervasive infrastructure. In the situation shown on the left, all these distracting interactions have to be taken care of by the user himself, since no personal Aura is involved to assist with this. On the right, the thick arrow is located between the Aura and the infrastructure. The user himself only has some distraction-free interactions with his Aura, while the Aura does all the hard work in setting up and managing the pervasive computing infrastructure. To accomplish this, an Aura needs to contain information about the personal preferences of the user, policies, and on-going tasks.[13]

Components

Three high-level components can be identified within project Aura. These are the

- the Task Manager;
- the Environment Manager, and;
- the Context Observer.

The Task Manager embodies the concept of the personal Aura. It consults the task information of the user and proactively requests the environment to set up the capabilities that support the tasks of the user. The Environment Manager is responsible for locating, setting up and managing computing capabilities and resources. The Context Observers keep track of changes in the physical context, so the Task Manager and Environment Manager can react to this. The technological contribution of project Aura can be found in these three main areas.

2.2 Standards for short-range wireless communication

Today the number of devices with some kind of computer chip embedded in them is countless. Imagine what it would be like to be able to harness all this processing power efficiently by letting all these devices work together. Although new developments in this particular area follow up on each other at a rapid pace, the full potential has yet to be reached. Thus it is not surprising that letting devices work together seems to be the new adage. The two projects that were discussed above are good examples of this. However, it would get a bit messy if we had to physically connect all these devices with wires and plugs. Fortunately there are several wireless protocols that enable us to communicate through air, without the need for cables, almost like real speech if one thinks about it. In this chapter we discuss three of the current standards for wireless communication.

2.2.1 Bluetooth

The core architecture for the Bluetooth protocol consists of an RF transceiver, a baseband, and a protocol stack. Figure 2-3 depicts this.

RF Transceiver

The transceiver operates in the globally unlicensed ISM band at 2.4 GHz. The bit rate is 1 Megabit per second and can be boosted to 2 or 3 Mb/s with Enhanced Data Rate [EDR]. The 79 channels in the band are ordered from channel number 0-78 and are spaced 1 MHz beginning at 2402 GHz. Bluetooth-enabled devices that are communicating share a radio channel and are synchronized to a common clock and frequency hopping pattern. Frequency hopping is used to make the protocol more robust to interference from other devices operating in the same band. The physical channel is sub-divided into time units known as slots. Data is transmitted between Bluetooth-enabled devices in packets. These packets are situated in the slots. Packets can fill one or more consecutive slots, allowing larger data chunks to be transmitted if the circumstances admit this.

Bluetooth is primarily designed for low power consumption and affordability and has a relatively short range (1, 10 or 100 meters). It makes use of low-cost transceiver microchips that are embedded in each device.

Class	Maximum Output Power	Range
Class 1	100 mW (20 dBm)	~ 100 meters
Class 2	2.5 mW (4 dBm)	~ 10 meters
Class 3	1 mW (0 dBm)	~ 1 meter

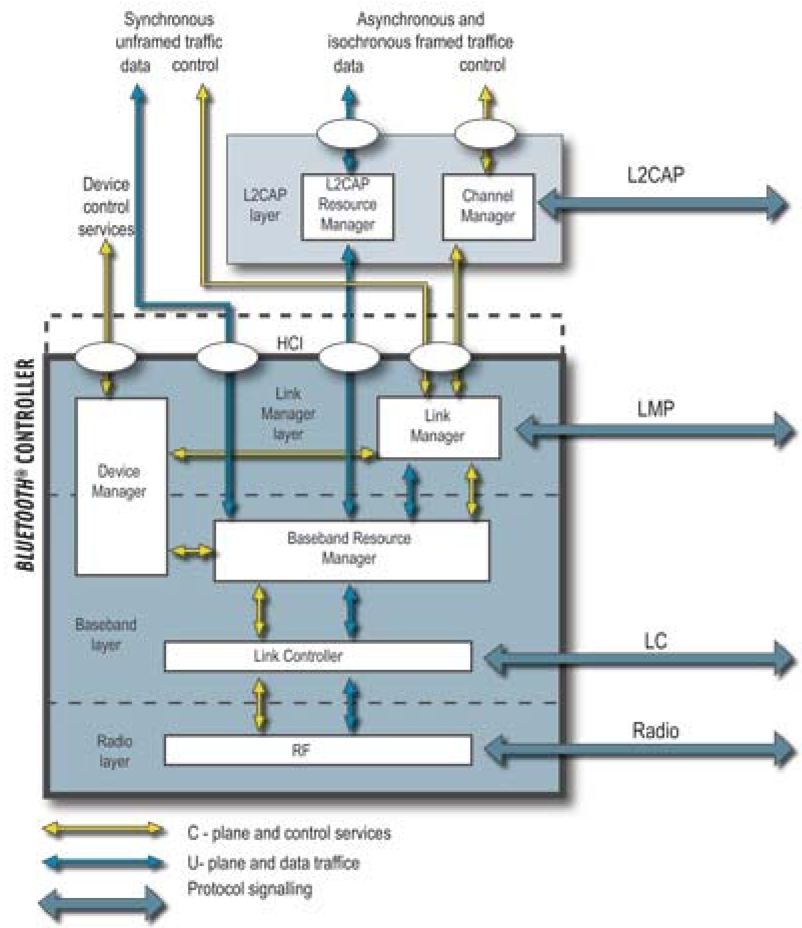


Figure 2-3: Bluetooth core system architecture

Baseband

The Bluetooth Baseband is the part of the Bluetooth system that specifies or implements the medium access and physical layer procedures between Bluetooth devices. Several devices can be joined together in what is called a piconet. One device owns the clock and the frequency hopping pattern and is called the master. All the other devices in the piconet are called slaves and have to synchronize themselves with the master. The frequency hopping pattern is a pseudo-random ordering of the 79 frequencies in the ISM band and is algorithmically derived from the clock of the master and certain fields in the Bluetooth specification address.

Two or more piconets can be joined in what is called a scatternet. To form a scatternet, some units, called gateways, belong to different piconets. Such a unit can be a slave unit in more than one piconet but can act as a master in only one. Besides this, it can transmit and receive data in only one piconet at a time. To visualize this, imagine the following. You are on the phone with a friend, using your Bluetooth headset, while at the same time you are uploading pictures from your computer to your phone. Your phone now acts as a gateway, being the master in the piconet with your headset and slave in the one with your computer.

Protocol Stack

Different protocols are used to set up communication between devices. The Link Manager Protocol [LMP], the Logical Link Control and Adaptation Protocol [L2CAP], and the Service Discovery Protocol [SDP] are required. Additionally, the Host Controller Interface [HCI] and Cable replacement protocol [RFCOMM] are almost universally available and almost always included. LMP is used for control of the radio link between two devices. L2CAP supports higher level protocol multiplexing, packet segmentation and reassembly, and the conveying of quality of service information. SDP is used to discover other devices to communicate with. HCI is a standard for communication between the host stack (e.g. a mobile phone) and the controller (the Bluetooth chip).

Versions

Versions 1.0 and 1.0B survived only shortly due to many problems, one being the difficulty for manufacturers to make their products interoperable. Version 1.1 was the first serious version of Bluetooth. Version 1.2 improved connection and discovery speed, and added Adaptive Frequency Hopping [AFH] as well as higher transmission speeds. In version 2.0 Enhanced Data Rate [EDR] was introduced for faster data transfer, rendering transmission speeds of up to 2.1 Mb/s to even 3 Mb/s. Besides this, EDR even provides for lower power consumption through a reduced duty cycle (the fraction of time that a system is in 'active' state). The latest version, 2.1, was released by SIG on July 26, 2007. It enables better filtering, lower power consumption when devices are in sniff low-power mode, encryption key refreshing and other security improvements. No major changes were made. This version is, just like the others, fully backward compatible with version 1.1.

Security

Currently there are three modes of security for Bluetooth access between two devices.

- Security Mode 1: non-secure;
- Security Mode 2: service level enforced security;
- Security Mode 3: link level enforced security.

Devices can be either trusted or untrusted. Security levels for services are divided into services that require authorization and authentication, services that require authentication only and services that are open to all devices.

All packets that are transmitted are encrypted. Encryption is based on a shared cryptographic key that relies on the Bluetooth PIN, which has been entered into one or both devices.

When talking about security related to Bluetooth, one inevitably comes across the terms Bluejacking and Bluesnarfing. Both are not to be confused however, since the first is legal whereas the second is not. Bluejacking is described as the sending of unsolicited messages over Bluetooth to Bluetooth-enabled devices such as mobile phones or PDA's. Essentially the name field is used to contain a message. This is possible, since the allowed length for a file name can be up to 255 characters, enough anyway to contain a small (personal) message. People who do not know what is going on usually think their phone or PDA is malfunctioning, which doubles the fun of course.

Bluesnarfing on the other hand is the unauthorized access of information from a wireless device through a Bluetooth connection. Bluesnarfing is typically used to obtain access to calendars, contact lists, e-mails and text messages on mobile phones. Just think of the contact lists of celebrities like Paris Hilton that pop up now and then on the internet. Usually, these celebrities have been bluesnarfed.

2.2.2 ZigBee

At first sight ZigBee[14] might look as a slower (concerning data throughput) version of the Bluetooth protocol, but in fact it was developed to serve very different applications, its main goal being to optimize power consumption. Besides this ZigBee is low-cost and aimed at applications that require a low data-rate, long battery life and secure networking.

ZigBee	Bluetooth
Very low duty cycle, very long primary battery life.	Moderate duty cycle, secondary battery lasts same as master.
Static and dynamic star and mesh networks, >65000 nodes available.	Quasi-static star network up to seven clients with ability to participate in more than one network.
Low latency.	Very high Quality of Service [QoS] and very low, guaranteed latency.

Direct Sequence Spread Spectrum allows devices to sleep without the requirement for close synchronization.	With the Frequency Hopping Spread Spectrum it is extremely difficult to create extended networks without large synchronization costs.
Ability to remain quiescent for long periods without communications.	Provides three modes of lower power consumption: sniff mode, hold mode and park mode.
Lower cost. The retail price of a ZigBee-compliant transceiver approaches $1.	Low cost. The retail price of a Bluetooth-compliant transceiver approaches $3.

On the official website a number of examples of applications are mentioned that would benefit from a ZigBee solution. These include Demand Response, Advanced Metering Infrastructure, Automatic Meter Reading, Lighting controls, HVAC control, Heating control, Environmental controls, Wireless smoke and CO detectors, Home security, Blind, drapery and shade controls, Medical sensing and monitoring, Universal Remote Control to a Set-top Box which includes Home Control and Industrial and building automation. Besides this, according to its developers ZigBee is the only wireless standards-based technology that is suited to fill the unique needs of remote monitoring & control and sensory network applications, enables deployment of wireless networks with low power, low cost solutions and is able to run for years on cheap batteries for a typical monitoring application.

Market Name Standard	ZigBee IEEE 802.15.4	GSM/GPRS CDMA/1xRTT	Wi-Fi IEEE 802.11b	Bluetooth IEEE 802.15.1
Application Focus	Monitoring & Control	Wide Area Voice & Data	Web, E-mail, Video	Cable Replacement
System Resources	4 KB – 32 KB	16 MB+	1 MB+	250 KB+
Battery Life (days)	100 – 1000+	1-7	0,5 – 5	1-7
Network Size	Unlimited ($2^{\wedge 64}$)	1	32	7

Maximum Data Rate (KB/s)	20 – 250	64 – 128+	11000+	720+
Transmission Range (meters)	1 – 100+	1000+	1-100+	1-100+
Success Metrics	Reliability, Power, Cost	Reach, Quality	Speed, Flexibility	Cost, Convenience

ZigBee outperforms other protocols if it comes to battery life. This is not entirely fair however, since the other protocols have primarily been targeted at rechargeable devices. ZigBee aims for those devices that do not use much power in the first place and have expected battery lives of up to ten years. Another advantage is the large networks that are supported and the relatively short (dis)connection period (~30ms) compared to Bluetooth (~2s). There are disadvantages though. Throughput is significantly less (250 KB/s) than that of Bluetooth (720+ KB/s), which poses a limitation to the possible applications of ZigBee.

2.2.3 Near Field Communication

Near Field Communication[17] [NFC] is a short-range high-frequency wireless communication technology which enables the exchange of data between devices over a distance of about 10 centimeters. Using the same technology as RFID it is mainly aimed at mobile phones. It can be used to transform the phone into a contactless card or to exchange information at close range. Applications may vary from mobile ticketing, and mobile payment to smart posters. With Bluetooth 2.1 it will also be possible to pair NFC devices with Bluetooth devices by putting them close together. This is supposed to save a great deal of time, since the process of activating Bluetooth on both sides, searching for other devices, waiting, pairing, and finally, authorizing is replaced by the process of putting the two phones next to each other.

Because of the close range however, NFC is practically suited only for pull marketing rather than push marketing, but it is an interesting new technology and the fact that the latest version of Bluetooth (2.1) includes it makes it worth to mention it here.

2.3 Bluetooth broadcasting

A new trend in pervasive computing that has been developing steadily for the last few years is proximity marketing (see paragraph 1.4 - Proximity marketing) via a Bluetooth broadcast. In this paragraph we discuss a number of projects (commercial and open source) to see what is currently happening in this field. Important factors in this discussion are the number of simultaneous connections, the range, different types of content distribution (text, images, audio, video, etcetera), and the class of Bluetooth antenna / transceiver used.

2.3.1 Commercial projects

We start with discussing commercial projects concerning Bluetooth broadcasting, starting with a leading authority in this field: BlueCasting from Filter WorldWide.

2.3.1.1 BlueCasting

BlueCasting[18] is a commercial product developed by Filter WorldWide (previously known as Filter UK). According to their website, BlueCasting 'enables brands, content owners and site owners to deliver mobile content to consumers backed by powerful campaign management and reporting tools'.

BlueCasting comes in four types: *Express*, *Event*, *Extreme* and *Kiosk*. *Express* is suited for the smaller locations such as bus shelters, phone boots, etcetera, *Event* is meant to be used for one-off events (music festivals, sport events, trade shows, etcetera) and is also supplied as battery-powered, *Extreme* deals with large spaces such as airports and uses a number of networked servers to function properly, and finally, *Kiosk* comes with an interactive touchscreen to allow direct user interaction.

BlueCasting allows for the following content to be distributed:

- text (as .txt files),
- still images (as GIF or JPG files),
- animated images (as Animated GIF files),
- audio (as WAV, RMF, MP3, MP4 or Ringtone files),
- video (as RM, 3GP or MP4 files),
- Java applications (as JAR files),
- vCard (Business Card files), and
- vCal (Calendar Event files).

Furthermore, the system uses Bluetooth Class 1 dongles and provides additional security in the form by requesting a 4 digit PIN number prior to delivering data to the target handset. If the BlueCast server identifies a device it starts delivering content based upon a set of rules defined by the owner. BlueCasting does not provide information about how many connections can be established at the same time.

2.3.2 MobiTouch

The MobiTouch Cube[19] comes with 4 integrated Bluetooth 2.1+EDR Class 1 antennas and can have a maximum of 28 parallel connections. This number of simultaneous connections can be increased to 56 by installing an external module. The range of the system is potentially 100 meters since MobiTouch uses Class 1 dongles, but as they also explain on their website Bluetooth is a two-way protocol and thus the maximum range also depends on the mobile phone model. They estimate the maximum range somewhere in between 30 and 35 meters in open space. It is possible to send different types of content, although the different types are not explicitly mentioned on the website. MP3, video, images and ringtones are hinted at though.

2.3.3 BlueSixty

Another player on the mobile marketing market is the company OneSixty BV with their product BlueSixty[20]. Just like BlueCasting they provide solutions for retail shops, events, restaurants and outdoor advertising. It is possible to send images, video, audio, text and links and games / applications. Just like BlueCasting BlueSixty uses Class 1 transmitters, allowing for a range up to 100 meters, depending on the receiving device. The number of simultaneous connections is 21. However, this number is extendable.

BTS100 Pocket

2.3.4 BLIP Systems

BLIP Systems[21] is another player in this crowded market and provides solutions for the same range of possibilities as BlueCasting and BlueSixty. They divided the categories into airport, cinema, mall, out-of-home, sports events and tourism. Their solution goes by the name of BlipZones and is presented as a complete package for professional mobile proximity marketing and information systems. It comes in three different modules:

- Basic,
- Competition, and
- BlipExplorer.

The Basic module is used for conventional push marketing and works with a central server accessible from any web browser. Via the web browser different campaigns can be configured. The Competition module allows for pull marketing by encouraging people to turn on Bluetooth in order to win prizes. BlipExplorer is a small application that users can install on their mobile phones. One of its applications distributes mobile brochures, which are updated seamlessly on the devices of the end-users on which BlipExplorer is installed.

The hardware unit used is BlipNode L2i, which is a small and intelligent access point connected to a central server. The central server ensures that users will not receive the same communication twice. It can be Powered over Ethernet (PoE) or use regular power supplies. The BlipNode is able to handle up to 21 connections simultaneously, because of the implementation of three Bluetooth 2.1 modules. This number of concurrent connections can be increased by deploying more BlipNodes in the same zone.

The BlipNodes connect to each other either via the built-in Ethernet port, via a 3rd party 3G USB dongle, or via a WiFi client. The BlipNodes are controlled and configured through the graphical interface called 'BlipManager'. The BlipManager is used for configuration, e.g. adjusting the range of the BlipNode or configuring the services of the access point.

2.3.5 Bloo2

Bloo2[22] sells PC marketing software for small businesses allowing them to create a Bluetooth marketing campaign. Besides the software they include a Calls 1 Bluetooth USB transmitter. They also sell Bluetooth access points which they call BluePods. These come in two different types. The first one supports up to 21 simultaneous downloads per Pod zone and can also communicate over Wi-Fi and 3G. The other pod only allows up to seven connections and supports direct cable connection only. They do not mention which class of Bluetooth transmitter they use for which of their products.

2.3.6 BlueBlitz

The Magic Beamer from BlueBlitz[23] can be used to distribute and receive mobile content for advertising campaigns. The Magic Beamer is sold in three different versions: small, big and multi-client. The big and multi-client versions already have three Bluetooth 2.0 EDR Class 1 dongles attached of which each dongle can handle a maximum of seven simultaneous connections. However, according to BlueBlitz it is possible to add as many dongles as you wish. The transfer rate of the dongles is 2.5 Mb/s. Content that can be transferred spans the domains of video, graphic, text, sound, and software.

2.3.7 BroadTooth

BroadTooth[24], developed by LondonDev, does not add anything new. The information on the product website is concise. Judging from their claim to be able to reach potential customers at about 30 meters they use a Class 1 transmitter. Nothing is mentioned about the number of simultaneous connections or different types of content that can be exchanged. They do mention a price however. The purchase of a one year license, including installation on-site and free support for one month, costs £ 1000.

2.3.8 Alterwave

Alterwave[25] refers to the Bluetooth access points as ‘hot spots’, which are managed by a central server just like most other companies. With respect to the content supported by Alterwave, they follow their competitors in providing support for the distribution of video, MP3, applications and Java games. Alterwave does not provide details concerning the class of the Bluetooth transceiver used and the number of simultaneous connections.

2.3.9 Other commercial projects

The commercial projects mentioned above are just the tip of an iceberg. Examples of other projects besides the ones mentioned in paragraph 1.4 - Proximity marketing are

- Bloozy (http://www.bloozy.co.uk),
- DMS Blue Media (http://dmsbluemedia.com),
- Punch Kick Interactive (http://www.punchkickinteractive.com/mobile/services/bluetooth-marketing.php),
- ProxiBlaster (http://www.proxiblaster.com/) and
- BluetoothOn (http://www.bluetoothon.com/).

2.3.10 Open Source projects

To the best of our knowledge there are no open source projects concerning Bluetooth broadcasting. This suggests that it still is a rather new field and little non-commercial research has been done on the subject.

2.4 Known issues regarding Bluetooth broadcasting

Advantages of broadcasting via Bluetooth are numerous. Nowadays, most people carry around some sort of Bluetooth-enabled device, such as a mobile phone, a PDA, or some other handheld. The costs for broadcasting are low and it is possible to distribute content that suits the location perfectly. However, the other side of the coin is that, despite the vast number of entrepreneurs in this field mentioned above, not everything about Bluetooth is as glorious as it seems. Just like every other technology, it has weaknesses too. Known issues are the different ways in which receiving devices handle the reception of the distributed content and the limit of only seven active simultaneous connections per Bluetooth dongle. Specifically related to this project are the issues that arise when trying to cover a large enough area, issues regarding the 'freshness' or 'up-to-date-ness' of the distributed message, and of course context awareness. In this paragraph we will discuss these issues in-depth.

2.4.1 Association with remote devices

Since Bluetooth is mainly used in mobile phones and other handheld devices there is no guarantee that every client will (be able to) receive the content as was intended. Even if we limit our scope to Bluetooth-enabled mobile phones only, the problem remains that these come in a vast variety of brands and models, each having its own way of dealing with content received via the Bluetooth protocol. Some phones even require you to set up a secure channel by means of entering a PIN before the transmission of data can be started.

To overcome this problem we have investigated Bluejacking as a possible solution. As explained above however, Bluejacking is more of a funny trick than a solution. It would severely limit the content that could be exchanged to simple text messages of a very short length. Even in the scope of our research this would not be sufficient.

2.4.2 Vertical scalability

Another type of issues regarding the Bluetooth protocol is vertical scalability, i.e. issues that have to do with the available bandwidth. The first limitation of Bluetooth is that it only allows up to seven active slaves in a piconet. Using park mode up to 255 devices can be supported. Problems occur when trying to deliver data to a large group of people (> 7 persons) who are on the move. Possible solutions can be divided into two groups: a) expanding the network by adding hardware to increase the number of simultaneous connections or using scatternets and b) using some kind of round robin scheduling for the active devices. The easiest solution of course is to add some extra dongles. Previous research[26], however, showed that Windows seems to be having problems supporting multiple dongles and it is not known in what way interference will play a role with respect to the number of dongles occupying the same zone. More on this will be discussed later on in this thesis. Then we will also see why scatternets are not a suitable option to overcome this problem. Should it be physically impossible to use more than one dongle in an acceptable way (which is highly doubtful with regard to most of the commercial projects that allow for 21 or sometimes even more simultaneous connections) we can turn to some kind of round robin scheduling algorithm as a final resort.

2.4.3 Horizontal scalability

Another challenge is to cover a large enough physical range. When covering larger areas such as a whole faculty, multiple antennas need to be set up at different locations. The challenge lies in the distribution of these antennas and in the handling of the gathered data. Should one use multiple servers distributed in a master – slave configuration or one central server? Will this be fast enough or do we need some kind of caching scheme to keep the system operating at a certain speed? We provide the answers to these questions further on in this thesis.

2.4.4 Freshness / context awareness

Finally, when broadcasting messages, the system needs to keep track of the messages that are sent in order to prevent spamming people with the same messages over and over again. Or there should be some kind of expiration date after which the message may be resent to users who have already received one. Besides this, the content of the message may be dependent on the location of the user. This means that the system needs to be context-aware. Given the context of this thesis, it is not unthinkable that a student moving around the Faculty of Computing Science would get different messages there than when he or she is wandering about the Faculty of Arts.

2.5 Other Bluetooth projects

Bluetooth is not limited to the domain of broadcasting content to clients. We have come a long way since Bluetooth was originally introduced as a cable replacement. By now Bluetooth is used in almost every domain where short-range wireless communication plays a role. Before discussing the main research conducted in this thesis, we mention very briefly what other kind of research projects involving Bluetooth in one way or the other have been done.

Murphy et al. [27] performed a feasibility study on using Bluetooth for short-term ad hoc connections between fast moving vehicles. They created a hardware testbed to make an empirical analysis of the time it takes to establish Bluetooth connections and the range at which these connections can be established. Their results are very encouraging. A vehicle traveling at 100 km/h is in range of the antenna for a staggering 18 seconds (provided that dongles of Class 1 are used). However, the measured maximum range for Bluetooth Class 1 and 2 seems a bit optimistic (250m for Class 1 and 122m for Class 2). By making small adjustments to the Bluetooth baseband protocol they also managed to reduce the discovery time of other Bluetooth-enabled devices.

Energy consumption is also an important issue regarding Bluetooth. Yan, Zhong, and Jha compared Bluetooth and Zigbee in terms of design cost, performance, and energy efficiency[28]. Another paper by Eliasson, Lundberg, and Lindgren discusses time synchronous sensor networks[29]. They suggest that communication delays and energy consumption can be optimized with the combination of clock synchronization and a time activation schedule.

Body Area Networks [BAN's] are also a popular subject of study. Nurmi et al.[30] describe how we can use a phone to retrieve information from a body area network and send this information to a server. This can be especially useful in a medical context (pervasive health care) where it can be used to monitor people. Mobile inter-body-area-networking is about connecting BAN's with each other to form a chain to pass on information to a server. This is applicable, for example, in the fields of sports and health care. In a paper from Lipphardt et al.[31] MarathonNet is introduced. The idea stems from the fact that athletes already wear sensors to gather data about their skills, but the trainer has to be close to read out the data or it has to be done afterwards. By using other BAN's to forward the information the reach can be extended and data analysis can be improved. In this paper they put the system to the test during a half marathon around a lake.

Lombriser et al.[32] discuss buttons with sensors that communicate with each other in order to gather information about the context of the person who is wearing them. As an example, they took an office worker and tried to recognize what he was doing (drinking water, moving a computer mouse, opening a drawer, etc.). Again, this could be very useful in a medical context, where it can be used to monitor suspicious behavior of the aged and ill.

Finally, many researchers focus on scatternets. Kettimuthu and Muthukrishnan have researched whether Bluetooth is suitable for large-scale sensor networks. They conclude that, at the moment, there are still scalability issues that have to be overcome when forming larger scatternets such as collisions and increased delays.[33]

3 Bluetooth broadcasting at R*u*G

In this thesis we investigate the feasibility of a distribution system that uses Bluetooth broadcasting to send relevant location-based information to students about exams, lectures and daily news. For a start the research has been limited to students at the faculty of Computing Science of the University of Groningen. The announcements for the pilot will consist of the announcements from the electronic learning environment at the University of Groningen that goes by the name of Nestor. Teachers use Nestor to give information about courses, to take electronic examinations, and to exchange documents within their own department. Students use Nestor to read announcements, to cooperate with group members, and to hand in assignments.

For this project the following requirements are set up:

- the communication protocol that is to be used must be Bluetooth,
- the system range must cover the Faculty of Science or more specific the Zernike complex, Bernouilliborg and Discovery Bus and
- the system must be able to distribute messages to large moving groups consisting of a more than 21 users with open Bluetooth devices.

This thesis focuses on the scalability of Bluetooth broadcasting which is mainly captured in the third requirement. To be able to determine if this project is feasible we need to know how a system that broadcasts messages via the Bluetooth protocol is able to cope with large mobile groups of people with Bluetooth-enabled devices. The fact that one Bluetooth antenna is only able to hold seven active connections is a known bottleneck for these kinds of systems. Given this information, the main research question will be the following:

'What are the hardware and software requirements to set up a scalable message distribution network to distribute Nestor announcements within the faculty of Computing Science to mobile phones of large moving groups of at least 28 students with Bluetooth-enabled devices using the Bluetooth communication protocol?'

This question can be used to determine the feasibility of this project and help us to answer the question of which type of Bluetooth broadcasting is to be recommended with respect to the final goal of the project, being to distribute context-aware information to students via the Bluetooth protocol.

3.1 Sub questions

In order to answer this question, some sub questions concerning related issues are formulated. First of all, we need to understand that the quantitative analysis of the scalability of the network is twofold. One issue involves the number of students that have to be reached at the same time, while the other one relates to the coverage of the network. Students must be able to receive announcements regardless of their position within the specified perimeter, in this case the faculty of Computing Science. We will discuss the first issue in this thesis. The coverage-related issue is left for further research.

Scalability in the context of this project means the possibility of the system to handle extra users without affecting its functionalities or performances or increasing the range of the network without considerable performance loss.

Because the number of connections for a Bluetooth access point is limited to 1 master connection with seven slaves, the first scalability issue concerns the number of connections that can be made at the same time using Bluetooth, or in other words:

'What are the possibilities to handle more than seven Bluetooth connections at the same time?'

This scalability issue is **user-based**. The other challenge related to coverage is **location-based**. However, this challenge falls beyond the scope of this thesis. Finally, considering the ultimate goal of the project, which is to set up a scalable system that is able to distribute context-aware information to users, some other questions are of importance. For instance:

- *'What is the average percentage of users who have Bluetooth activated by default?'*
- *'What are the possibilities to increase this number?'*
- *'Is it more desirable to use a pull- instead of a push-based marketing strategy?'*
- *'Is there a limitation to the number of dongles that can be used for broadcasting?'*

These questions, along with the main research question of course, are answered in this thesis. Besides these questions a lot of other interesting questions come to mind. What is the minimum number of access points needed to cover a certain area? How does the network deal with possible interference with other applications / networks? What is the usability of the system? Performance is also important. The system has to be reliable, easy to understand, and has to take into consideration that its users are on the move, which means it has to be fast. These questions fall outside the scope of this research, but are interesting for future research.

The goal of the project is to actually deploy a prototype of the system described above and evaluate it by testing in real-time. Simulation can be used for those cases where the number of users with Bluetooth-enabled devices or the area that has to be covered grows too large to test in real life.

3.2 Research methodology and thesis content

In this paragraph we describe the global setup of this thesis. A short overview of the following chapters – each with a small description added – is given here. Part II of this thesis starts with an overview of the architecture (Chapter 4) of the project. We first describe the project as a whole. After this we fill in the global components to get a specific mapping for the part of the project this thesis is about. In Chapter 5 we focus on the implementation for this specific part of the project. We discuss our software, used libraries, and important design decisions. The experimental setup as well as a description of the tests we performed is given in Chapter 6. The purpose of this chapter is to describe the metrics used and to enable the reader to redo all the tests. The objective data which form the results of the tests are presented in Chapter 7. Various tables and figures are used to represent the data in a more comprehensible way. Our discussion of the results and the main conclusions of this thesis are provided in Chapter 8. We will go back to the research question and its derivatives to see if they are answered in a satisfying way. Finally, our research ends with giving suggestions for further research in Chapter 9, based on the discussion of the results of this research. Anomalies encountered during our research as well as topics that were beyond the scope of this thesis but are interesting to investigate nonetheless, are suggested here.

Part II Architectural overview

4 Architecture

4.1 Global architecture

The ultimate goal of this research is to investigate the feasibility of a scalable system that is able to distribute context-aware content to (possibly) large mobile groups via the Bluetooth wireless communication protocol. In order to achieve this we need:

- a server with a Bluetooth Access Point,
- a group of people with Bluetooth-enabled devices, and
- an external database storing the content that needs to be distributed.

For the global architecture we also assume that we need more than one access point to cover enough range. This leads to some extra requirements:

- multiple servers with Bluetooth Access Points and
- an internal database to store logs and information for the different servers to be able to work together.

Figure 4-1 shows how the different components are working together. The servers equipped with Bluetooth Access Points retrieve the content they distribute from the external database. The Bluetooth Access Point is responsible for discovering Bluetooth-enabled devices in range and establishing a connection over which the content may be distributed. Each access point may have connections with multiple Bluetooth-enabled devices. The internal database stores the logs and overhead for the Access Points to work together successfully.

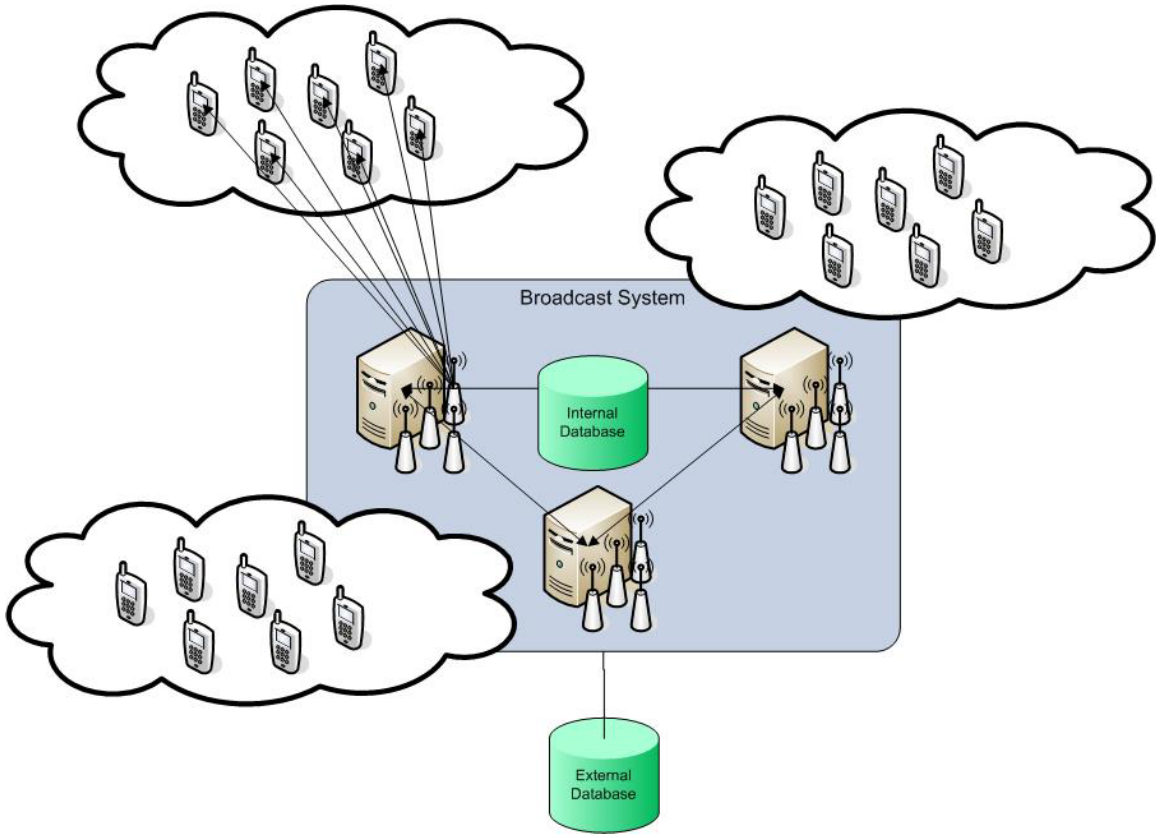


Figure 4-1: Global architecture

4.2 Project architecture

The architecture described in the previous paragraph is still pretty general. For this project we can fill in the global architecture of Figure 4-1 with more detail. Since one of the main ideas of the project is to be able to keep students informed regarding their current colloquium, the external database will be the database of the Nestor Digital Blackboard, storing announcements for different courses. The announcements can be distributed through a feed as XML over HTTP. This is a fact based on previous research done regarding this subject. The internal database can be a simple MySQL database. Other types databases could also be used, but we chose MySQL because it is free and we did not expect the amount of data traffic to be so high that we should need a heavier database system.

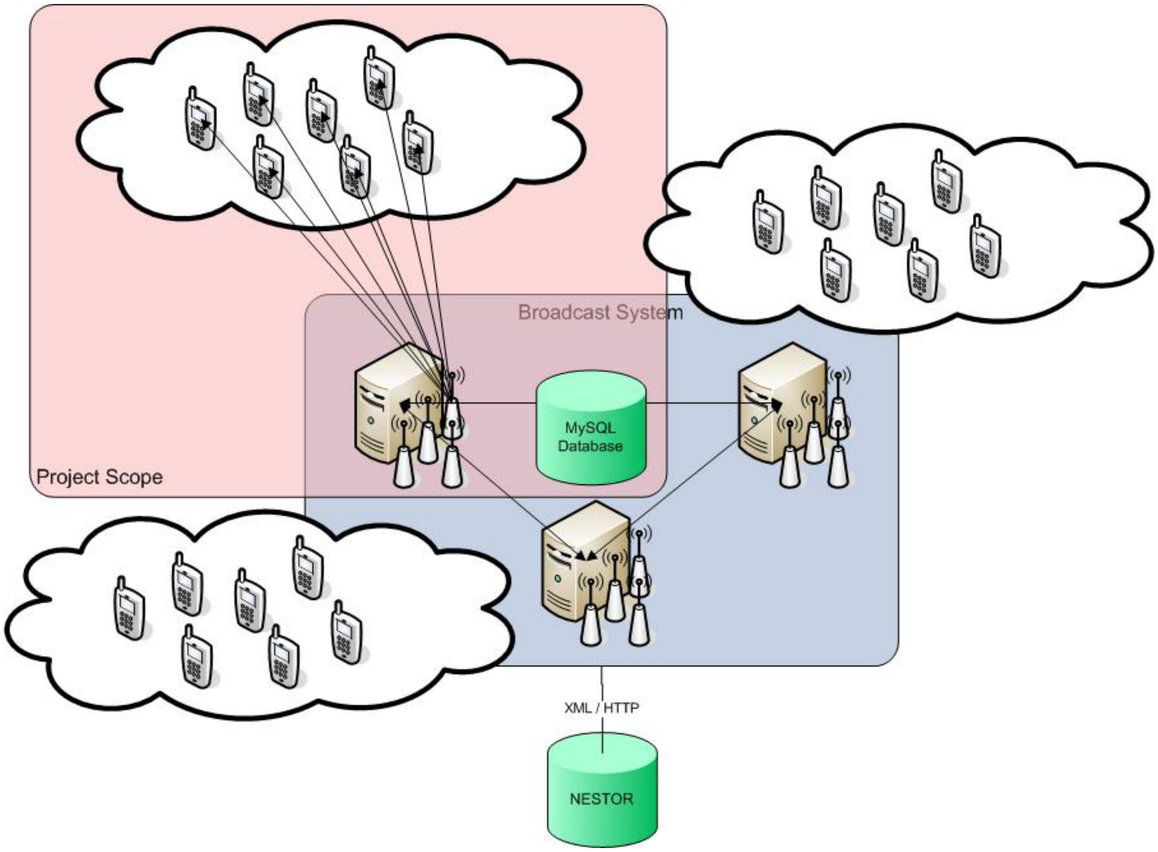


Figure 4-2: Project architecture

This thesis focuses on the scalability of the system with respect to the size of the group with Bluetooth-enabled devices. This means the components of interest to us are the (mobile) group with Bluetooth-enabled devices, the server running our software, the number of access points for that server, and finally, the internal database to log the results of our tests. Expanding the system to a multi-server environment and the retrieval of content for distribution from external databases is beyond the scope of this thesis.

5 Implementation

Figure 5-1: The system in action

5.1 Goals and design decisions

We first describe our goals for the software and then discuss each design decision based on these goals in the chronological order they were made.

5.1.1 Goals

The goals of our implementation were derived from the main research question of this thesis. Basically these goals boil down to the implementation of a system that

- can handle more than seven simultaneous Bluetooth connections,
- is able to send simple text messages to Bluetooth-enabled devices via the Bluetooth protocol, and
- preferably (for sake of scalability) is able to work with more than one Bluetooth dongle.

These goals had a large impact on the technical possibilities. We found out that to accomplish these goals we were forced to make some important design decisions.

5.1.2 Choice of operating system

The first decision we had to make deals with which operating system would be best to use. Each operating system supports a different Bluetooth stack (cf. Table 5-1). These different Bluetooth stacks are the software implementation of the definitions of the Bluetooth protocol stack. An important factor that influenced this decision was whether or not the stack supports multiple dongles.

Operating system	Best supported Bluetooth stacks
Microsoft Windows	Widcomm Microsoft Windows BlueSoleil
Mac OS	Max OS X
Linux / Unix	BlueZ

We quickly found out that this choice was made for us, since all Microsoft Windows stacks and the MAC OS X stack do not support multiple dongles. BlueZ, the stack for Linux / Unix, does however, so this was the logical choice. From all different Linux distributions we first chose to use Ubuntu 8.04, a stable version of this operating system. For the last two endurance tests we switched to Ubuntu 9.04. The fact that Ubuntu is simple and accessible tipped the balance in favor of this distribution of the Linux operating system.

5.1.3 Choice of hardware

The next step in the process was to choose the appropriate hardware. Since we had to do tests on different locations, we decided to use a laptop to facilitate us in this mobility. Because the operating system (Linux Ubuntu version 8.04 and later on version 9.04) in combination with our software would not require many resources, any laptop with a reasonable amount of memory and CPU processing power would do. The final tests were performed on a DELL Inspiron 5150 after a DELL Inspiron 1150 proved to be too lightweight.

Since the laptop we used had only two USB ports, we bought a USB-hub to be able to use enough dongles. We chose to use a 10-port Sitecom CN-052 USB-hub, widely available at in consumer electronics stores.

As for the actual Bluetooth interfaces, better known as Bluetooth dongles or Bluetooth adapters, we tested a number of different brands (cf. Table 5-2). When we discovered the Sitecom CN-523 did not run stable we switched it for its predecessor, the Sitecom CN-521, a much more stable dongle.

	Brand	Specification	Class	Price
	Linksys	Bluetooth USB Adapter Class 1 USBBT100 ver. 2 Bluetooth 1.1, USB 1.1 Max data speed 721 Kbps	1	€ 29,99
	Sweex	Bluetooth 2.0 Class 1 Adapter USB Bt211 Bluetooth 2.0 EDR Max data speed 3 Mbps	1	€ 9,58
	Sitecom	Bluetooth 2.0 USB Micro Adapter CN-523 Bluetooth 2.0 Max data speed 3 Mbps	1	€ 9,95
	Sitecom	Bluetooth 2.0 USB Adapter CN-521 Bluetooth 2.0 Max data speed 3 Mbps	1	€ 9,99
	Conceptronic	2.0 USB Adapter 200m C04-104 Bluetooth 2.0, USB 1.1 Max data speed 3 Mbps	1	€ 14,99

	MSI	Bluetooth USB Dongle BToes Bluetooth 1.2, USB 1.1 Max data speed 723 Kbps	2	n.a.

5.1.4 Choice of programming language and libraries

Our choice for the programming language was based on a combination of factors. We chose Java for its known portability features and the greater number of available Bluetooth libraries. Besides this the B2Build project, on which this project is based, was also written in Java.

Company Name	javax.bluetooth Support	javax.obex Support	Java Platforms	Operating Systems	Price
Avetana	Yes	Yes	J2SE	Win-32, Max OS X, Linux Pocket PC	€ 25 , Free
Blue Cove	Yes	Yes	J2SE	Win-32, Max OS X, Linux Pocket PC	Free (LGPL)
Electric Blue	Yes	Yes	J2SE	WinXP SP2	$15 USD
Harald	No	No	Any platform that supports javax.comm	Many	Free
JavaBluetooth.org	Yes	No	Any platform that	Many	Free

			supports javax.comm		
Rococo	Yes	Yes	J2ME, J2SE	Linux, Palm OS	€ 2500 , Free

We chose Bluecove[34] as our library based on the fact of its support for multiple platforms, its good documentation and – maybe just as important – the fact that is was free to use under Lesser General Public License [LGPL].

Finally our choice of database to store the logs of our software fell on MySQL. For this we installed an Apache 2.0 Webserver with a MySQL database and a PHPMyAdmin administrative panel. At the beginning of this chapter the final system is shown in action (cf. Figure 5-1).

5.2 Components

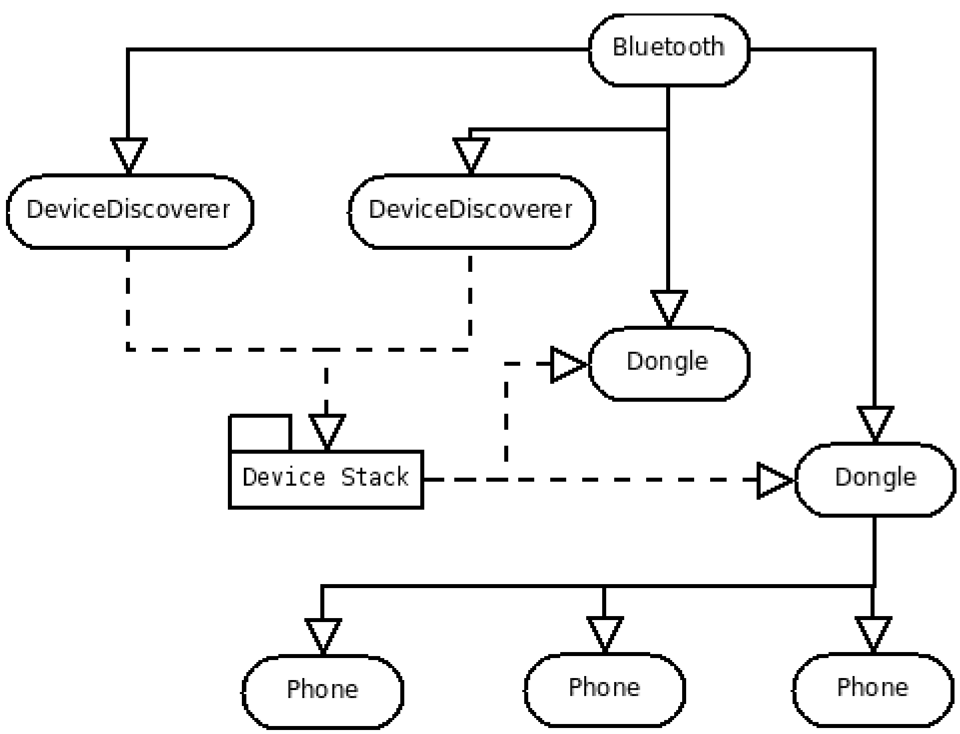


Figure 5-2: Global structure of the software

The global structure of the software is depicted in Figure 5-2. It shows how the different components are working together. The main components shown in the figure (Bluetooth, DeviceSearcher, Dongle, and Phone) will be discussed (in this order) next.

5.2.1 Bluetooth.java

Bluetooth.java contains the main module of the software. This class is used to setup the right configuration to start the broadcast. Based on the command line parameters it starts the right number of different threads that do the actual work. First the DeviceSearcher threads are started, followed by the Dongle threads. The Bluetooth class also initializes the stack that is used to store all waiting devices and a method that can be used to retrieve waiting devices from the stack.

5.2.2 DeviceSearcher.java

DeviceSearcher threads are always trying to discover new Bluetooth-enabled devices in range. To ensure the search is done as efficient as possible, the threads run asynchronously. The time difference is based on the discovery cycle and the number of threads running. For example, if there are two threads running and the discovery cycle lasts for 10 seconds, the threads are started 10 / 2 = 5 seconds after each other. All devices that are discovered by each of the DeviceSearcher threads are added to the Device Stack.

5.2.3 Dongle.java

Simultaneous to the DeviceSearcher threads, the Dongle threads are continuously checking the Device Stack for new devices added. To minimize the chance of one thread trying to serve everything on the stack and cause exceptions, these threads also run asynchronously. Once a new device is found, the Dongle thread retrieves it from the Device stack and launches a Phone thread, which is used for the final part.

5.2.4 Phone.java

The Phone threads, which are launched by one of the Dongle threads, are used to do the actual work. First the thread tries to discover the right service for the Bluetooth-enabled device. Once the right service is found, an attempt to open an OBEX connection is made. Once the connection is established the message is sent over this connection.

Part IV
Evaluation

6 Experimental setup

We set up several experiments to test the current limits of the Bluetooth technology and our implementation with respect to Bluetooth broadcasting. Five different tests were performed. In order of discussion the tests measured the range of the system, the reception of the message, the responsiveness of Bluetooth broadcasting, its scalability with respect to the number of simultaneous connections and finally the endurance of the system during a multiple hour run.

6.1 Range test

First of all we are interested in the maximum distance at which a Bluetooth-enabled device (in our case a mobile phone) can be found and communicated with. For this test we will use two different classes of Bluetooth USB adapters, i.e. one Class 1 dongle and one Class 2 dongle. According to the specifications devices of the first class type have a range of approximately 100 meters, whereas the range of devices belonging to the second class reaches no further than about 10 meters. However, it seems that the effective range of Class 2 devices is extended when connecting it to a Class 1 receiver. Reasons for this are the higher sensitivity and transmission power of Class 1 devices.

As with almost all our tests different factors play a role when trying to determine the maximum range of our system. First of all there is difference between an indoor and an outdoor environment. The second important factor is the brand of Bluetooth dongle that is used. As with all products different manufacturers have different quality standards. Thirdly we need to consider possible interference when working with multiple dongles at the same time. Finally the class of the transmitting Bluetooth dongles is of importance. According to the specifications Class 1 has a greater range than Class 2 dongles. This is why the range test was split up into two separate tests. The first test was to investigate the difference in range between an indoor environment and an outdoor environment. The second test was performed to find out the variations in range when using different brands of dongles and to measure the impact on the range when using multiple dongles.

6.1.1 Indoor versus outdoor

For the first experiment two series of three tests were performed. One series was performed in open air (near the Bernoulliborg) whereas the other series was conducted inside the Bernoulliborg. To measure the difference in sensitivity between a Class 1 and a Class 2 type of transmitter, the experiment was performed twice, one time for each class of transmitter. The temperature outside was -1 °C as opposed to about 21 °C (room temperature) inside the Bernoulliborg. For a Class 1 dongle we used the Linksys and for Class 2 the MSI (cf. Table 5-2). The messages were sent using the B2Build software implemented by Simon Dalmolen and Jasper Hafkenscheidt in the predecessor of this project. The distance inside was measured in meters with aid of a scaled map of the building. The distance outside was calculated with Google Maps by choosing recognizable spots and we counted our steps. We are aware that these are not the most precise methods, but for this test a deviation of one or two meters does not matter. The goal of this range experiment was to get an impression of the range of the different types of dongles in different environments (indoor and outdoor) and to determine which type of dongles were to be preferred for the final project. Furthermore this experiment allowed us to get a global impression of the number of transmitters that are needed to cover the whole faculty. The Bluetooth-enabled mobile phones used for this test were a Samsung Omnia and a Samsung SGH-E900 mobile phone.

6.1.2 Different brands and interference

Since the first test was performed using only one dongle, we performed another test later on in the project to find out if working with multiple dongles would cause interference to have an impact on the range at which devices could be found. We only used Class 1 transmitter dongles for this test and also picked different brands (Linksys, Sweex, Sitecom, and Conceptronic) to measure a possible difference in quality between the various manufacturers of Bluetooth dongles. To minimize the chance of other devices interfering with the dongles, we performed this test in open air. To measure the range, the distance from the sender to the receiver was counted in steps. The results of the Sweex dongle were used as a benchmark from which to calculate the ranges for the other manufacturers. Each measurement was done twice. First the discovery range was determined for each of the dongles and after this the maximum range at which a simple text message could be received, was measured. This test was done during a relatively sunny day (18 °C) at a deserted location in an attempt to minimize interference. The mobile phone used for the test was a Sony Ericsson K610im.

6.2 Reception test

Given the heterogeneity of implementations of the Bluetooth stack on devices, it is also interesting to test how the same message is received on different devices and if there are limitations regarding size (number of characters). After all, since our goal is to develop a system that is able to distribute Nestor announcements, it would not harm us to find out if it is possible to exchange messages of similar size in the first place. Another question is to what extent we can 'control' the reception of the message, in this case with the meaning of bypassing user interaction. Bluejacking now immediately comes back into the picture. But how is such a 'bluejacked' message displayed? And how is the user notified of an incoming message? Each brand of mobile phone has its own way of storing Bluetooth messages and notifying the user. Users that are not made aware of receiving a message will be blocking the channel. Does the time you have to accept an incoming message depend on the brand of phone or is this something we can control? What are other possibilities here?

For this test we tried to send messages to different brands and different models of mobile phones. We looked at the following characteristics: if the message is delivered, how it is delivered (where is stored, how is it displayed) and if and what kind of agreement is necessary from the user. The test was performed with the same software (B2Build) as the first range test (indoor vs. outdoor). Besides a description we also took screenshots of the display of the mobile device in order to give a better impression of how and where the message was received.

6.3 Responsiveness test

The speed at which Bluetooth-enabled devices that come in range are able to find each other and exchange a message is an important factor for broadcasting among mobile devices. If it takes too long to discover a Bluetooth-enabled device, we might 'miss' some devices capable of receiving messages, since the person carrying it could have walked out of range before his device was discovered or when an attempt was finally made to send the actual message. Transmission of the messages should also be fast due to similar reasons.

We measure the responsiveness of our system in the classical way of considering the time difference between the initial transmission of a message and its total reception (thus merging the contributions of latency and bandwidth). This time delta is measured in seconds and considered infinite if the message cannot be delivered. Failure in delivery can be caused by a multitude of factors. The device may not be found although in range, the device may need to be paired first, or there may be some other problem with the connection resulting in delivery failure. We also expect responsiveness to be closely related to the number of simultaneous connections at a given moment. The more simultaneous connections the system establishes the more clients can be served at the same time. Of course the impact on bandwidth when sending multiple messages simultaneously has to be taken into account.

To circumvent the difficulties that testing with a mobile group would unavoidably raise we came up with an alternative setting. We rephrased the main question for this test as follows:

'*How many messages are successfully delivered to a group of n persons with Bluetooth-enabled devices in t seconds?*

Given the range of our system (a radius of about 30 meters) and the average walking speed of a person (about 4 km per hour or 67 meters per minute) we calculated that a device would be in range for about one minute or 60 seconds on average. This question was put to the test during a bachelor course at the University of Groningen with 50 students present, 37 of which had a Bluetooth-enabled device. The test was split up into four rounds, each round using a different amount of discovery and delivery dongles. During the first round one dongle was used for discovery and one for delivery, in the second one for discovery and two for delivery, then one for discovery and three for delivery and finally two for discovery and two for delivery. Four Class 1 dongles were used for this test, two of them from Conceptronic, one from Sitecom (CN-523) and one from Linksys. During the first three test rounds the dongle from Sitecom was used as the discovery dongle. During the fourth and final round both the dongle from Sitecom and one of the dongles from Conceptronic were used as discovery dongles. Results for the test were collected through a questionnaire. For more information about the questionnaire see Appendix A – Questionnaire.

6.4 Simultaneous connections test

The maximum number of active connections we can serve per dongle is seven. Fortunately, the number of active connections is not the maximum number of connections one dongle can hold. Other devices (up to 255) can be inactive or parked, waiting for the master device to activate them at any given time. Usually this is done in a round-robin fashion. To efficiently serve large groups of mobile users it is paramount to break this limit of seven active connections. There are two options. The first and preferred method is to be able to plug in multiple dongles into one server, thereby scaling up to seven extra users every time a new dongle is plugged in. The advantages of this method are that all users can be served at the same time and at a busy day (e.g. a high school excursion) the system can be scaled up by just plugging in some extra dongles, something almost everybody will be able to manage. Not all Bluetooth stack implementations allow this however. For a start, the Bluetooth stack that is used by Microsoft Windows and the one that is used by Apple do not support multiple dongles (cf. Table 5-1). Fortunately BlueZ, the implementation of the Bluetooth stack for Linux, does support multiple dongles.

The test to determine the maximum number of simultaneous connections was performed simultaneously with the responsiveness test and later on simultaneously with the endurance test. The software was implemented to log the maximum number of active simultaneous connections along with some other variables during each session.

6.5 Endurance test

Since all other tests only had the system running for a short period of time, we made the choice to perform another final test where the system would run for a much longer period. No signs or other forms of communication were put up to make people aware of our test, allowing us to gather some reliable statistics about Bluetooth broadcasting in practice. We performed three tests during three consecutive nights (Thursday, Friday and Saturday) in one of the most popular streets (Peperstraat) for going out in Groningen. For the first test two discovery dongles and three delivery dongles were used (two Conceptronic and three Sitecom CN-521 dongles). The second test and third test were executed with two discovery dongles and four delivery dongles (two Conceptronic and four Sitecom CN-521 dongles). The Sitecom dongles were used as discovery dongles in all three tests. After the test on Friday night the OS on the laptop was updated from Ubuntu version 8.04 to Ubuntu version 9.04. During the first test the system ran from 18:30h until 2:30h. The second test was performed between 22:45h and 18:00h and the third test from 18:30h until 12:00h. One small modification was made for the third test with respect to the other two: the name of the sender, the not so encouraging 'bluetooth_laptop_1', was replaced with 'Priscilla'.

7 Results

7.1 Range test

Since the range test was split up into two separate tests the results of these tests are also presented separately.

7.1.1 Indoor versus outdoor

The first range test was meant to provide information about the difference in range between indoor and outdoor broadcasting and the difference between the use of Class 1 and Class 2 Bluetooth dongles. The results for this test are listed below in Table 7-1.

Location	Test round	Class 1		Class 2	
		Omnia	*SGH-E900*	*Omnia*	*SGH-E900*
Open air	Round 1	43 meters	47 meters	7.5 meters	13 meters
	Round 2	40 meters	48 meters	7.5 meters	11 meters
	Round 3	43 meters	48 meters	7.5 meters	14 meters
Inside Bernoulliborg	Round 1	28 meters	30 meters	9.5 meters	15 meters
	Round 2	28 meters	31 meters	10 meters	14 meters
	Round 3	27 meters	30 meters	10 meters	15 meters

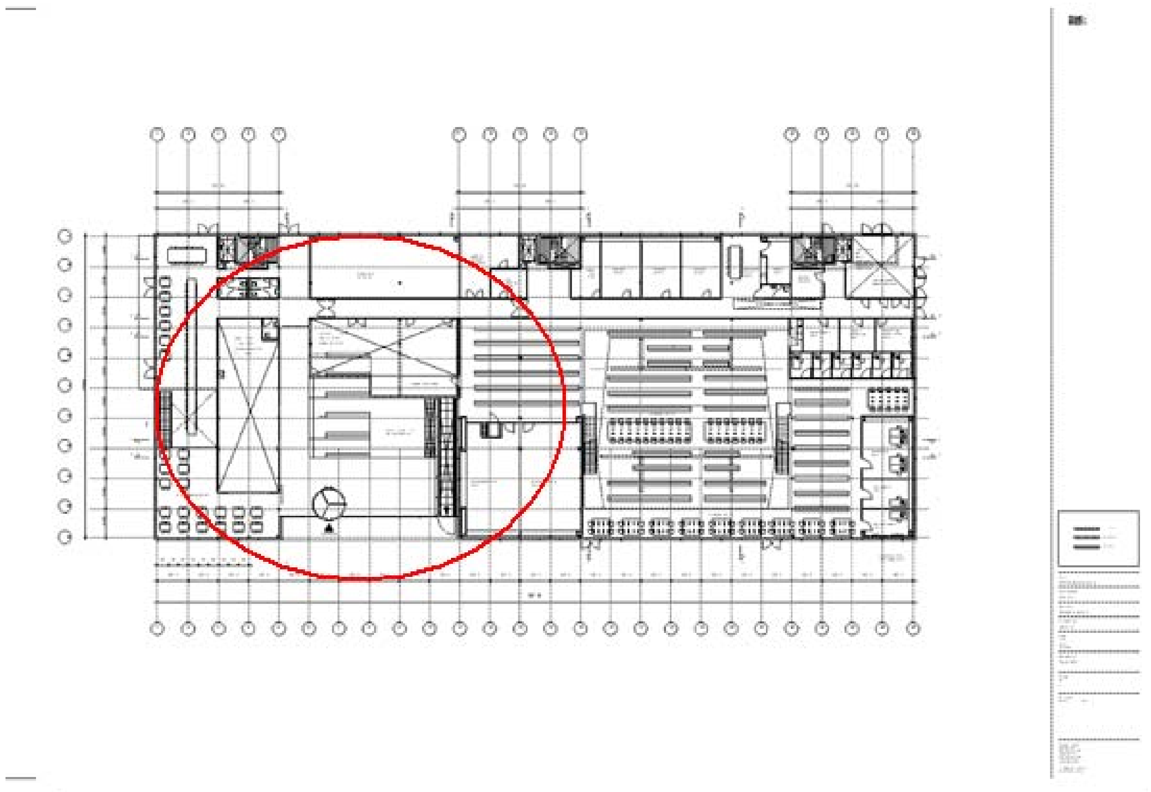


Figure 7-1: Theoretical maximum range within the Bernoulliborg (ground floor)

The data from Table 7-1 is visualized in Figure 7-1. It shows the theoretical radius based upon our measurements with the Class 1 dongle. However, it was very difficult to determine an exact range area due to many kinds of obstacles. Furthermore the radio waves did not propagate outside the Bernoulliborg. An attempt to transmit a message from inside the building to a destination outside the building over less than 5 meters with as only obstruction a glass window failed hopelessly. Finally, although the representation of the range in Figure 7-1 is given in two dimensions, the radius of the Bluetooth antenna needs to be visualized as a sphere.

7.1.2 Different brands and interference

The second part of the range test involved testing with different brands of dongles and multiple dongles to determine the effect of interference and difference in product. The results for this test are listed below in Table 7-2. The first MR-column contains no values, since our software needs at least two dongles to be able to send messages.

# Dongles	1		2		3		4	
Brand	DR[2]	MR[3]	DR	MR	DR	MR	DR	MR
Linksys	110 m	-	110 m	27 m	110 m	27 m	111 m	29 m
	110 m	-	110 m	27 m	110 m	27 m	108 m	25 m
Sweex	30 m	-	31 m	31 m	36 m	36 m	32 m	32 m
	30 m	-	31 m	-	33 m	33 m	35 m	35 m
Sitecom	45 m	-	45 m	32 m	46 m	14 m	44 m	10 m
	45 m	-	45 m	32 m	46 m	14 m	46 m	11 m
Conceptronic	45 m	-	45 m	33 m	45 m	33 m	45 m	32 m
	45 m	-	45 m	33 m	45 m	34 m	45 m	33 m

7.2 Reception test

To test the reception of a simple text message via Bluetooth on a mobile phone, a small sample of phones has been put to the test in order to find out the differences in reception among different brands and models of phones. Besides textual data (cf. Table 7-3) screenshots of the displays of the devices have been taken to visualize the handling of the message by different phones (cf. Figure 7-2).

Brand	Model	Message	Remarks
Nokia	6210	Successfully delivered.	Displayed as text message in inbox.
	6300	Successfully delivered.	Displayed as text message in inbox.

[2] DR = Distance Range; the maximum range at which the test device has been found

[3] MR = Message Rang; the maximum range at which the test device was able to receive a message

	E71	Successfully delivered.	Displayed as notice.
Samsung	SGH-E770	Not delivered.	Asked for PIN-code.
	SGH-E900	Not delivered.	Asked for PIN-code.
	Omnia	Successfully delivered.	Displayed as text message.
Sony Ericsson	K770i	Successfully delivered.	Displayed as webpage.
	W710i	Successfully delivered.	Displayed as webpage.
	W810i	Successfully delivered.	Displayed as webpage.
Sharp	920SH	Not delivered.	Asked for PIN-code.

As can be seen, different brands of mobile phones each have their own way of handling the acceptance of the incoming message, storing it, and presenting it to the user.

(a)
Nokia – View of message

(b)
Nokia E71 – Message received in text message inbox

(c)
Sony Ericsson k770i – Message saved on memory stick

(d) Sharp 920SH – Pairing request

(e) Sony Ericsson w710i – Accept incoming message

(f) Nokia E66 – Accept incoming message

Figure 7-2: Reception of message by different brands of mobile phones

7.3 Responsiveness test

The responsiveness test was performed during a bachelor course with 50 students present of which 37 had a Bluetooth-enabled device. Results were gathered by means of a questionnaire and by logging data into a MySQL database. The results of the questionnaire are listed first, the results from the database second.

7.3.1 Questionnaire

There were 50 students present during the test. Of these 50 students there were 37 which brought a Bluetooth-enabled mobile phone or another device that was capable of receiving a simple text message via the Bluetooth protocol. The first test round was done using one discovery dongle and one delivery dongle. The next two rounds an extra delivery dongle was added each round. The fourth test round we switched to two discovery dongles and two delivery dongles. The percentage of messages that were received, according to the questionnaire, within a specific amount of time, during each test round is depicted in Figure 7-3.

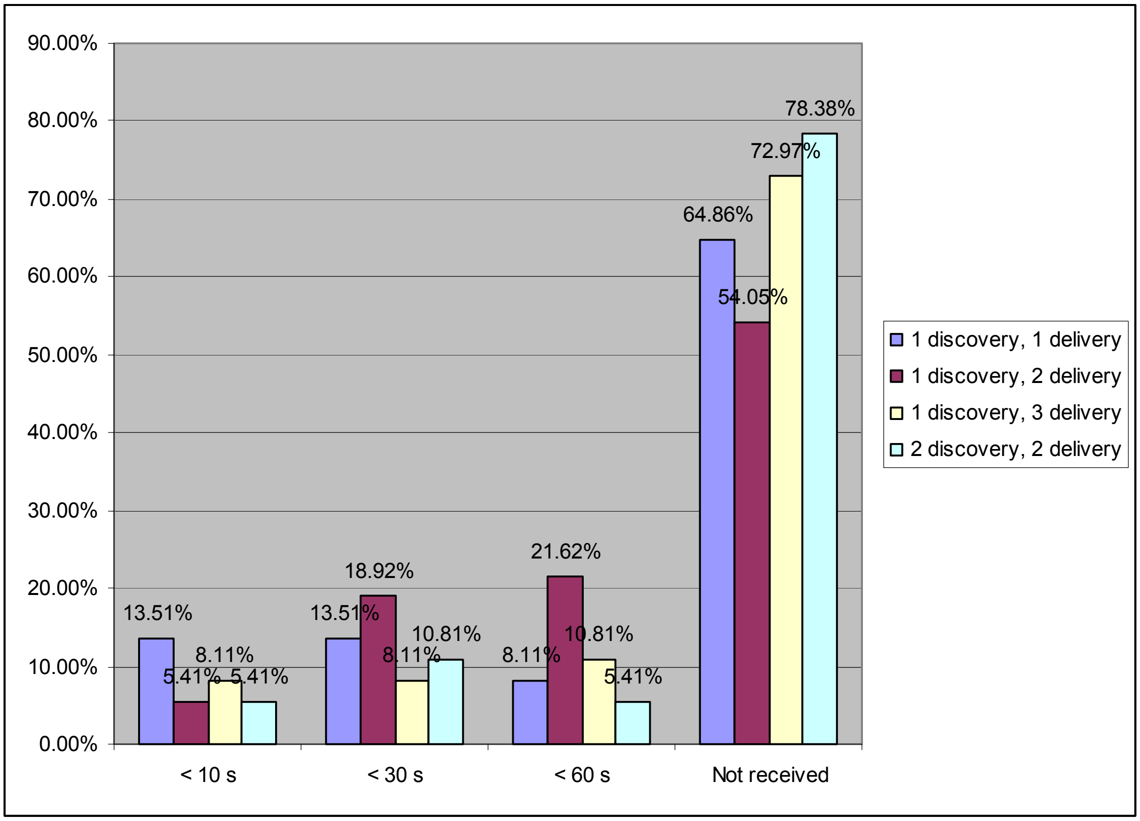


Figure 7-3: Percentage of messages received with varying number of dongles

Table 7-4 lists the actual number of messages that were received during each of the test rounds based on the results of the questionnaire. From left to right the table shows the test round, how many messages were received within 10 seconds, within 30 seconds, within 60 seconds, the sum of these results, and finally, the number of messages that were not received. The sum of the number of received messages and the number of not-received messages should add up to 37, since this is the number of participants with Bluetooth-enabled devices that engaged in the test. The number between parentheses indicates the number of messages that were successfully received, but which the test subject was unable to locate on his or her device. The absence of a number between parentheses means that all messages that were received, were also retrieved by the test subject.

Round	< 10 s	< 30 s	< 60 s	Received	Not received
1	5	5	3 (1)	13 (1)	24
2	3	3	4 (2)	10 (2)	27
3	2 (1)	7 (1)	8 (2)	17 (4)	20
4	2 (2)	4 (1)	2 (1)	8 (4)	29
total	12 (3)	19 (2)	17 (6)	48 (13)	100

Figure 7-4 depicts the relation between the successful reception of messages and the brand of Bluetooth-enabled device. Manufacturer Apple is listed in the legend of the chart, but not visible since all attempts to send a message to an Apple iPhone failed. The same holds for the one student who brought an NTC Touch HD for the first three rounds. The fourth round the message was delivered properly as can be seen in the chart shown in Figure 7-4. In total six different brands of devices participated in the test: Nokia, Samsung, Sony Ericsson, LG, Apple and NTC.

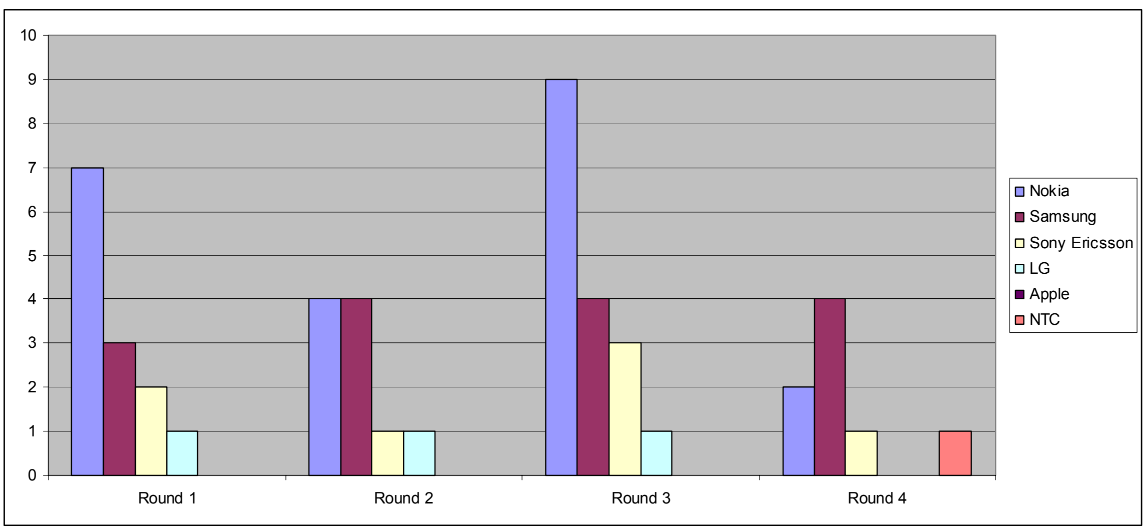


Figure 7-4: Number of received messages per round per brand

Since the devices that have successfully received a message vary per test round, we are also interested in the number of devices that successfully received a text message during at least one of the test rounds. It shows from Figure 7-5 that with 28 out of the 37 devices this has been the case. Just nine devices did not receive a single message.

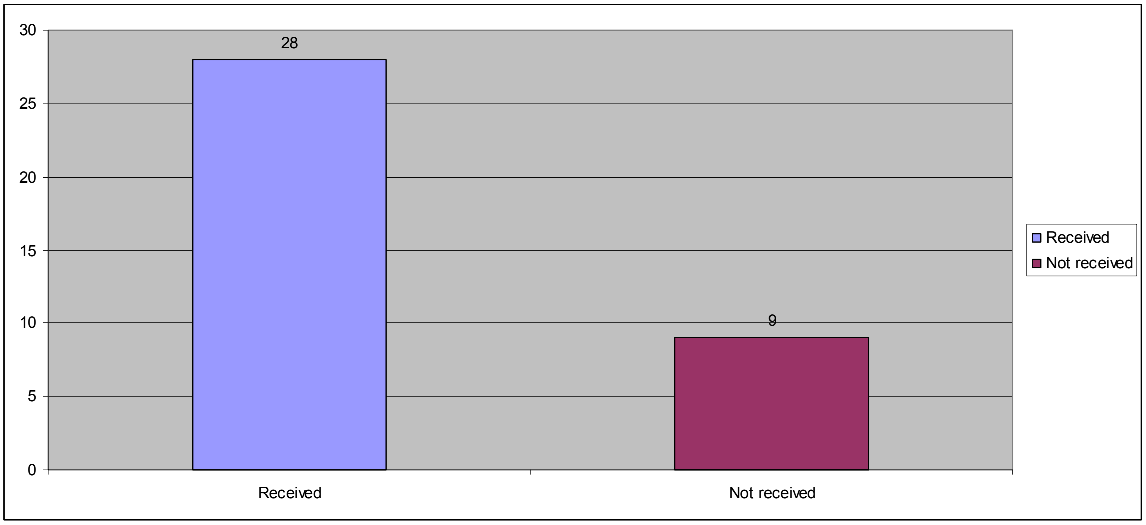


Figure 7-5: Devices that received a message during at least one of the test rounds

Finally, the total number of messages received per device after four test rounds is illustrated by Figure 7-6. For example, there were eight unique devices which had received two messages after four test rounds. It also shows that no device managed to receive a message all four tries.

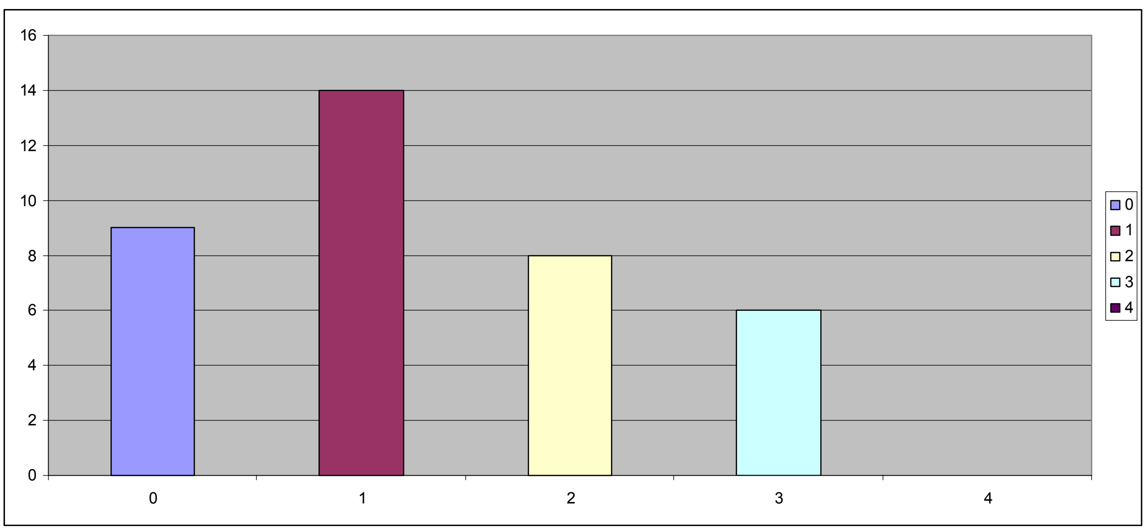


Figure 7-6: Total number of messages received by device after four test rounds

The next paragraph focuses on the data acquired from the MySQL database log.

7.3.2 MySQL database log

Our second series of results comprise the data collected from the MySQL database to which the software wrote its logs. If everything went according to plan, the data should closely match the results from the questionnaire. There is some room for discrepancies, but more on this can be found in the next chapter. Since radio waves are known to be able to propagate through walls and other solid objects it is interesting to find out how many distinct Bluetooth-enabled devices were found during each of the test rounds and how many unique devices were found in total given the four test rounds. Besides this it is also of interest to see how many attempts were made to send the message and how many messages were successfully delivered according to the database log of the software. The software also logged the number of times a connection was terminated and the number of times that no services were found on the device. Table 7-5 lists this information. In the columns that are divided into two the left number represents the number of times a certain action has happened whereas the right number represents the number of unique devices for which this has happened. For example, if we look at row one of Table 7-5, the right most column (no services found), we find out that during round one the software was not able to find any services in 12 cases. Of these 12 cases where no services were found, 10 unique devices for which this happened were involved.

Round	Unique devices	Delivery attempts		Successful attempts	Connection terminated		No services found	
1	36	28	20	14	2	1	12	10
2	36	42	24	9	2	1	31	20
3	34	43	30	17	2	2	24	16
4	38	63	26	8	4	3	51	20
total	40[4]	176	100	48	10	7	118	66

[4] The total number of distinct devices is not just a sum of the number of distinct devices during each round, since there is an overlap between the distinct devices found during each test round.

7.4 Simultaneous connections test

The measurements for the maximum number of simultaneous connections during each of the responsiveness test sessions were gathered and stored in a MySQL database. The results presented in this paragraph were extracted automatically by an algorithm implemented in a small script written in PHP. The MySQL log consists of two tables. One table stores the devices found by the discovery dongles and the other one stores the activities of the delivery dongles. Every time a connection is initiated or terminated (successful delivery, no services found, or another form of termination) this is logged into the database. The maximum number of simultaneous connections during each test round can now easily be calculated by traversing the result data, adding 1 for each newly initiated connection and subtracting 1 for each terminated connection. The results of this algorithm are listed in Table 7-6 below.

Round	# Delivery dongles	Max. possible # simultaneous connections	Max. # simultaneous connections reached
1	1	7	7
2	2	14	13
3	3	21	14
4	2	14	14

Later on in the project, the software was adapted so it could keep track of the maximum number of simultaneous connections by itself. This modified version of the software was used for the endurance test. Results are shown in the table below.

Round	# Delivery dongles	Max. possible # simultaneous connections	Max. # simultaneous connections reached
1	3	21	21
2	4	28	19
3	4	28	22

7.5 Endurance test

Each endurance test had a different length. The duration of the first test was eight hours, the second test lasted for about 19 hours and the third test took 17.5 hours. Results are presented in the table below (cf. Table 7-8). From left to right the columns list the test round, how long the software ran (i.e. the duration of the test), how many unique devices were found, how many times the software tried to send a message (delivery attempts), how many of these attempts were successful, how many times the connection was terminated prematurely and finally how many times the software failed to open a connection (no services found). Whenever a column is divided into two, the left box lists the total number of events whereas the right box lists the number of unique devices for which this event occurred. In total 47 messages were successfully delivered after a running period of 44.5 hours. This means an average of slightly more than one (1.06) successful delivery per hour.

Round	Running time	Unique devices	Delivery attempts		Successful attempts	Connection terminated		No services found	
1	8h	439	241	136	8	233	129	1567	412
2	19h	612	228	140	13	215	127	5314	606
3	17.5	883	568	288	26	542	270	8286	872
total	44.5h	1934	1037	564	47	990	526	15167	1890

8 Discussion

8.1 Range test

Before we discuss the results of the range tests we need to state something about the use of Bluetooth transmitters and receivers of different classes. First of all the theoretical range of a Class 1 Bluetooth dongle is said to be about 100 meters. Class 2 is designed to be able to transmit over distances up to 10 meters. Of course this is the range for two dongles of the same class communicating with each other. When mixing the classes the theoretical range of the Class 2 dongle is increased, because of the sensitivity and transmission power of the Class 1 transmitter.

8.1.1 Indoor versus outdoor

Testing with the Class 1 transmitter did not yield surprising results. The Samsung SGH-E900 proved to be a little bit more sensitive, but the difference was only a couple of meters. In open air the Samsung Omnia reached an average range of 42 meters with the Class 1 dongle. For the Samsung SGH-E900 the average distance covered was 47.5 meters. As expected (due to more interference) the range dropped when we tested inside the Bernoulliborg. The Samsung Omnia was only able to receive messages at an average distance of 27.5 meters whereas the Samsung SGH-E900 responded to the Class 1 transmitter up to about 30 meters away. Since the mobile phones we used for test were equipped with Class 2 Bluetooth receivers and we used a Class 1 transmitter, 30 meters was about what we expected.

At first, testing with the Class 2 transmitter went as expected. The Samsung Omnia constantly measured 7.5 meters. This is not too far, but expected and acceptable, for the theoretical range of a Class 2 transmitter is about 10 meters. The Samsung SGH-E900 performed a lot better with an average of 12.5 meters. Probably this phone has a more powerful / sensitive Bluetooth chip. However, when measuring inside the Bernoulliborg we found surprising results. The range *inside* the building appeared bigger than outside. About 10 meters for the Samsung Omnia opposed to about 14,5 meters for the Samsung SGH-E900. This was quite unexpected and we do not have a solid explanation for this.

Robert K. Morrow[35] writes that low power indoor wireless protocols such as Bluetooth are destined to encounter several impediments when trying to establish a link within suitable range. The first reason mentioned for this is that the indoor environment is usually quite cluttered. Every obstacle between the transmitter and the receiver translates into loss of signal strength. Besides this, walls, doors, and furniture can cause severe reflections of the transmitted signal, which means that multiple copies of the signal, with different phase relationships and different time delays, can arrive at the receiver. Another important aspect is the indoor competition for bandwidth in the 2.4 GHz spectrum. This competition can be intense and the result hereof is often severe interference that further reduces the reliable range. According to the literature the range inside the building should have been less than the range outside.

Since our test results prove otherwise, we have tried to come up with an explanation. However, the only possible significant difference that we can think of is that it was freezing outside when we did the tests, but it seems fairly unlikely that this would have influenced the results. We have found no research whatsoever describing the impact of temperature on the range of Bluetooth transmitters. Besides this the Class 1 transmitter did behave as expected, even in the cold. Finally we have not been able to find out about similar cases, a fact that labels this case an anomaly, since we are not able to explain it.

Finally, we have to mention that the range measured in the Bernoulliborg varied due to interference. Walls, windows, etcetera, showed to have an impact on the signal. During one of our tests we tried to send a message across a short distance (about 3 meters). The only difference with the normal experiment was that the receiving device was separated from the transmitting device by a window. With the window open, both Class 1 and Class 2 transmitters were able to get the message successfully delivered. Closing the window resulted in failure of deliverance for the Class 2 transmitter. The Class 1 transmitter however was still able to get a successful result. Nevertheless, the range decreased significantly. Walking a short distance (1 or 2 meters) away from the window already resulted in failure. We also were not able to get a message through to the classroom right above us.

8.1.2 Different brands and interference

When it comes to discovering nearby Bluetooth-enabled devices the dongle from Linksys outperforms the other ones by a landslide. The dongle must be very sensitive for it was able to find our test device up to a staggering maximum of 111 meters. In comparison, the next best dongle (Sitecom) only managed to find the test device at a respectable distance of 46 meters. Looking at the results of Table 7-2 in general the average discovery range seems to lie somewhere between 30 and 45 meters, which is as expected.

It becomes interesting when we try to send a message to the devices that are discovered. Apparently the discovery range has little to do with the range at which can get a message across. With the exception of the Sweex dongle all other dongles showed a significantly decrease in range when successfully trying to send a message. Adding extra transmitters did not bother most of the brands. Only the dongle from Sitecom had a decreased message delivery range when working with three or four transmitters. The dongle from Sitecom was by far the smallest of the dongles we used for the test. Perhaps its size made it more susceptible for interference. However, we have not been able to find evidence for a relation between the size of a Bluetooth dongle and its performance. Overall the dongles from Conceptronic proved to be the most stable and reliable.

8.2 Reception test

As expected the messages sent via the Bluetooth protocol are handled differently by the various kinds of mobile phones that are around. Some phones require you to set up a secure connection by pairing them with the transmitter before you are able to send anything to them via Bluetooth (cf. Figure 7-2d). This category will not be able to receive messages once the project is finished, since it is not possible to let the server guess the PIN-code used for the pairing process which is required to establish the private connection.

Most Nokia phones treat messages received via Bluetooth as normal text messages (cf. Figure 7-2a). They appear in the text message inbox along with the other text messages (cf. Figure 7-2b). Other phones display the message as a webpage or as a notice. For some phones, e.g. Sony Ericsson, we were not able to retrieve the message after we had received and read it. It seemed the message had mysteriously vanished from the phone or was at least stored at a location that was very hard to find. The fact is, however, that most Bluetooth-enabled mobile phones are able to receive the messages and – with a few exceptions – these messages can easily be retrieved.

Trying to control reception by using ‘bluejacking’ tested negative. The mysteries surrounding it quickly vaporized when it turned out Bluejacking was nothing more than misusing the field for the filename as the message. Besides the fact that it put a limitation to the number of characters that could be used for the message, the way the 'message' was displayed was not desirable at all. Suppose we try to send a message with name 'Hello this is a message' to someone. The receiver would then get a notification on his phone which would look something like 'Do you want to receive "Hello this is a message"?'. Clearly this is not what we are looking for in this project.

Another interesting fact is that some phones do not respond when receiving an incoming request for receiving a Bluetooth message. They do not vibrate or make any other audible sound. This means that it is very probable that some people will not notice the incoming message and will be blocking the channel. Unfortunately this problem is unavoidable and will have an impact on the performance of the software. When a user forgets to accept the message, there is no other option than to wait until a timeout event is generated. By default the connection will be terminated after 15 seconds. Of course it is possible to alter this value, but the question remains whether this is desirable or not. We think the focus should lie on visual ways to make users aware they are in a zone where they can receive messages via Bluetooth. Putting up posters and other visual information will encourage users to check their mobile phones. It is no coincidence that almost every commercial vendor of equipment for proximity marketing gives this advice to their customers.

8.3 Responsiveness test

We learn from Table 7-4 that the configuration in which we made use of one discovery dongle and three delivery dongles (test round 3) is the most successful. The number of actual messages delivered is 17 or 45.95%, which is not too bad, considering the software only ran for one minute. By using pull marketing techniques (let the user come to you instead of the other way around) the time that a client is in range could be increased, which will most likely have a positive effect on the number of messages delivered. Table 7-5 shows that there is a relation between the number of delivery dongles and the number of attempts to deliver a message. The more delivery dongles the system uses, the more delivery attempts are made. Adding an extra discovery dongle led to the discovery of more unique devices. 38 devices were found, which is one more than the actual number of participating devices (37), indicating the infiltration of a rogue device.

The results of the responsiveness test give rise to a number of other questions as well. First we can wonder whether the number of received messages increases when extra delivery dongles are added to the system. From analysis of the data it appears that this relation is not that obvious. Although the maximum number of successfully delivered messages was achieved during the third test round, there is an unexpected decrease during the second round and fourth round with comparison to the first round. Considering the fact that the number of delivery attempts to unique devices was higher, more messages should have come through. Why the data proves otherwise is difficult to explain. Increased interference from the fact of using more dongles could be an answer, but first of all this is hard to measure and secondly the software did not appear to have this problem during the third round, which had even more dongles running with possible interference and almost the same amount of delivery attempts (43 compared to 42) to even more unique devices (30 compared to 24). Despite all these apparent disadvantages 17 messages were delivered during the third round compared to only nine messages during the second round and eight during the fourth round. During our own test phase we already encountered a similar problem. When we ran the software several times in sequence, sometimes the test device would receive the message instantly, whereas at another time it would take a couple of tries before the software was able to find a service for opening a connection to the device. This seemed to happen at random as we were not able to discern a pattern in this behavior. Besides this the dongles seemed to cache a lot more information than appeared at first sight. This was discovered when we used another Bluetooth manager. Finally it turned out that turning Bluetooth off and on would sometimes have a positive impact on the discovery of a device. An explanation could be that the connection with a device from a previous test round is not terminated in the right way and the device still thinks that it is connected somehow.

To consider the success of broadcasting another question concerns which percentage of the test devices did not receive a message every single test round. This question is of importance, since we know there is a (probably significant) number of devices which are not able to receive a message at all. For instance, devices that require pairing fall under this category. When broadcasting you probably want to ensure that at least the majority of potential clients can be served. During our experiment only nine of the 37 devices did not receive a message every single round, which means 28 devices or 76% were able to receive a message: an encouraging result. This result is based on one group of test subjects, however, and it should be interesting to find out this number for other groups. For example, only two devices required pairing during our test. Both happened to be the same brand and model: Samsung SGH-E900. This number could be higher for other groups. Another factor is the fact the software only ran for one minute. Should users be successfully encouraged to stay in range for a longer period of time, the number of successfully delivered messages is very likely to increase.

Finally a remarkable fact is that during the second test round ten people indicated to have received a message. However, the log from the software shows us another number: nine messages were delivered successfully. We suspect that someone made a small error when filling in the questionnaire. Two out of the ten messages that were received were lost. Probably one of these messages was never delivered at all, but apparently one person thought it was. Based on the furthermore exact match of the log results to the results of the questionnaire we vote in favor of the log.

8.4 Simultaneous connections test

The maximum number of simultaneous connections was measured during two separate tests: the responsiveness test and the endurance test. During the responsiveness test a specially designed algorithm was used to calculate the maximum number of simultaneous during each test round. During the second test the software was adapted so that it was able to keep track of the number at run-time.

We see that during the responsiveness test for the configuration with one delivery dongle this number is seven, with two it is 13 respectively 14 and with three it is 14 (cf. Table 7-6). The fact that during round three only 14 out of a maximum of 21 simultaneous connections was reached is explainable. Although we did use three delivery dongles for this round allowing 21 simultaneous connections, this also implies that the speed at which the dongles handled devices could have been higher. The fact that the limit was pushed to the maximum during the fourth round, but not during the second round can be explained by the fact that two discovery dongles were used. With devices being found at a more rapid pace, it is not unlikely for the delivery dongles to have experienced a higher workload than when they were working with only one discovery dongle.

The number of simultaneous connections during the endurance test indicated that it had been three busy nights. During the first night (Thursday, also known in Groningen as Student's Night) the system was pushed to the maximum and served 21 simultaneous connections at its peak. The second night was an extra delivery dongle was added. The maximum number of simultaneous connections now dropped to 19. This could be due to the fact that more dongles decreased the workload per dongle or that Friday night was less busy than the night before. On Saturday the system passed the boundary of 21 serving up to 22 simultaneous connections at its peak.

8.5 Endurance test

The endurance test yielded some interesting results. The first night eight devices out of 439 received a message which means a success rate of 1.8%. The second night this percentage increased to 2.1% only to reach a staggering 2.9% the final night. If we consider the statistics from paragraph 1.4.3, which tell us to expect a success rate of about 1.25%, we can see that our system performed rather well. We can only guess what the number of successful deliveries would have been if we had put up posters to encourage people to turn on their Bluetooth. But all in all the results for the endurance test are more than acceptable and according to our expectations. We suspect that the use of a friendly sender name ('Priscilla' instead of 'Bluetooth_laptop_1') may have had a positive influence too. Based on the statistic that 90% of all people have Bluetooth turned off, this means that about 4390 people the first night, 6120 the second and 8830 people the third night were in range of our system during its run. Since the Peperstraat is one of the most popular places for going out these numbers are not unlikely.

8.6 Conclusions

Having discussed the results from all tests we are finally able to answer the main research question of this thesis and the questions that were derived from it. As we recall from Chapter 3 the main research question was formulated as follows:

'What are the hardware and software requirements to set up a scalable message distribution network to distribute Nestor announcements within the faculty of Computing Science to mobile phones of large moving groups of at least 28 students with Bluetooth-enabled devices using the Bluetooth communication protocol?'

Keywords here are the term scalable, defined here as the ability of the system to cope with large moving groups of at least 28 persons with Bluetooth-enabled devices, the location, which is the faculty of Computing Science, and the content that is to be distributed, i.e. Nestor announcements. Based on the literature study performed in this thesis and the results of our real-time testing we can draw the following conclusions. First of all, looking at the commercial activity during the last couple of years in the field of Bluetooth broadcasting in the form of proximity marketing, broadcasting content via Bluetooth seems to be a very lucrative business. Usually this means the product works, otherwise simple economic laws would have made it to disappear from the market.

Secondly the majority of users with Bluetooth-enabled devices have Bluetooth deactivated by default. This number is even estimated at 90% (see paragraph 1.4.3) which answers the following sub question:

'What is the average percentage of users which has Bluetooth activated by default?'

One of the cornerstones of successful broadcasting is encouraging users to turn Bluetooth on. Usually this is done via posters or other visual media, which answers another sub question:

'What are the possibilities to increase this number?'

A third consideration is that content that is distributed commercially spans a wide range: from simple text messages to business cards or video. From our own test we learned that most Bluetooth-enabled mobile phones are capable of receiving test messages and displaying them in a satisfying way. Although the location where the message is stored may vary, most of the times it is easy to retrieve it and view it. Since Nestor announcements are simple text messages, we can conclude that this poses no problem for our system.

Fourth, scalability is another thing. The larger the number of Bluetooth-enabled devices, the longer it takes the system to serve them all. Our best actual test result, 45.95% of the messages delivered successfully, is not bad, but not very encouraging either. Nevertheless, there are ways to improve this number. The success rate could be increased by using a different strategy. Instead of push marketing, the technique used with spamming, it is probably better to choose a strategy based on pull marketing. We expect that putting up large posters to make everybody aware of the system, encouraging them to turn on their Bluetooth and to wait until they received a message, would bring about a boost to the success rate of the system. Some commercial vendors take the pull marketing strategy even further by letting the users also be the ones initiating the exchange. Posters encourage the users to send a message via Bluetooth to the server in order to establish a connection over which various types of content can be received. Encouraging users to stay in range for as long as possible is important, since our tests showed that opening an OBEX-connection to send the message was highly error-prone. Sometimes we got lucky and a connection was opened on the first try. A lot of other times, however, the system was not able to find the right services and had to try again and again. The results of the endurance test tell us that for 439, 612 and 883 unique devices, the number of times no services were found are respectively 1567, 5314 and 8286 times. This means that in the worst case the system has to try to connect 10 times before establishing a connection successfully. Given enough time however, the system finds the appropriate service most of the times, but there is a good chance that users that are unaware of the system have walked out of range before this happens. This means that based on this information our answer to the following sub question

'Is it more desirable to use a pull- instead of a push-based marketing strategy?'

would be 'yes, it is probably more desirable to use a pull based marketing strategy'.

Considering the size of the group of people and the speed at which the group is moving scalability has its limits. Successful message delivery can be achieved by plugging in extra transmitters. This restricts the use of OS to Linux however. Windows does not allow more than one dongle to be active at the same time. The same goes for Mac OS X. The only solution we read about for this problem was using VMWare to simulate multiple machines, each controlling one dongle. In our software, one dongle is always used for discovery. Since one discovery dongle is perfectly able to find devices quickly, we think this method is faster than letting all dongles alternate between roaming for devices and sending them messages. This also answers the following sub questions:

'Is there a limitation to the number of dongles that can be used for broadcasting?'

and

'What are the possibilities to handle more than seven Bluetooth connections at the same time?'

The answer is using software running on Linux. Linux in combination with the BlueCove library supports multiple dongles. The main advantage is that scaling the system up becomes as simple as plugging in an extra dongle. We discovered that plugging in a large number of dongles (> 5) may cause some instability. This instability also depends on the hardware and distribution of the operating system. The software ran much more stable on a desktop than it did on the laptop. On the desktop we did not encounter a limit for the number of dongles. However, we stopped at 10, since this was the maximum number of slots our USB-hub provided. Upgrading the laptop to Ubuntu version 9.04 also gave more stability. The laptop we used was a Dell Inspiron, a model known to have some difficulties with the Linux OS.

Revisiting the main research question, we still need to give an indication of the hardware and software requirements for the system described in it. Basically all you need is a normal computer with Linux and Java running on it, an internet connection and at least two USB ports to plug in the Bluetooth dongles. The internet connection would be needed to retrieve the Nestor announcements and user information (supposing users need to register themselves in order to use the system). It is preferable to use a wired internet connection to minimize interference of the Wi-Fi protocol with the Bluetooth protocol.

Finally, our main conclusion is that broadcasting via the Bluetooth protocol to moving subjects at a fast pace is difficult, because there are so many factors to be reckoned with: interference, time constraints, opening a stable connection to a device, limitations of the operating system, the wide variety of end users, etcetera. However, with a few minor changes, using a pull instead of a push strategy for example, the performance of the system could be increased to be very acceptable indeed.

9 Suggestions for further Research

During the course of the project we came across some interesting questions which were beyond the scope of this thesis and we encountered some strange anomalies, which are in need of further research in order to find an explanation for them. In this chapter we summarize all these cases.

9.1 Interference

One of the first difficulties encountered during this project was the impact of interference. Bluetooth operates in the unlicensed 2.4 GHz band, which means it has to compete for bandwidth with other wireless communication protocols such as Wi-Fi. Besides this, we experienced an anomaly when testing inside the Bernoulliborg with the Class 2 dongles. An explanation could be that, at the time of the test, there was more interference outside. However, we did not have the equipment to measure this. Measuring the level of interference also becomes interesting if the system is expanded to cover more range by using more servers spread across the faculty. Does interference play a role for areas that are in range of more than one access point? It is obvious that more research is needed here.

9.2 Horizontal scalability

The next step in the evolution of this project is to increase the coverage. We used one access point, but to cover the whole faculty we are going to need more than that. This brings up a number of questions just for a start:

- What is the minimum number of access points needed to cover a certain area?
- What is the best server setup? Should they be independent of each other or should we use one master server?
- What is the best way to handle the handoff of a device when it moves into range of another access point?
- What are the impacts on scalability of the system?

Besides these questions, a lot of other ones will probably arise during research. To answer them, further research could concern deploying this system within the faculty of Computing Science. An attempt could be made in to cover the whole faculty, but covering just the ground floor would probably do, too.

9.3 Hard- and software limitations

How great an idea you have might be, sometimes you will find out that some things just are not possible. In our case, early in the project, we came across the problem of working with multiple dongles. In the case of Microsoft Windows, working with multiple dongles at the same time is simply not supported. The only solution for this was to simulate multiple machines with the aid of VMWare. It could be interesting to do more research to find out how stable this is. Besides this it would be useful to find out if there exists a maximum to the number of dongles that can be used simultaneously under Linux. We stopped our test at 10 dongles, but this was because our USB-hub was full. We did not look extensively into this, so further research on this subject is advisable, too, if only to investigate the increase in the level of interference.

9.4 Performance

The fact that we performed one responsiveness test with only 37 Bluetooth-enabled devices calls out for a new performance test in order to get stronger data. To push stress testing to a maximum, larger groups with Bluetooth-enabled devices should be tested. We tested with 37 devices, but how does the system cope with 50, 75, or 100 devices? How does the system perform when confronted with these numbers of people? More test data would also let us draw stronger conclusions regarding the feasibility of the project.

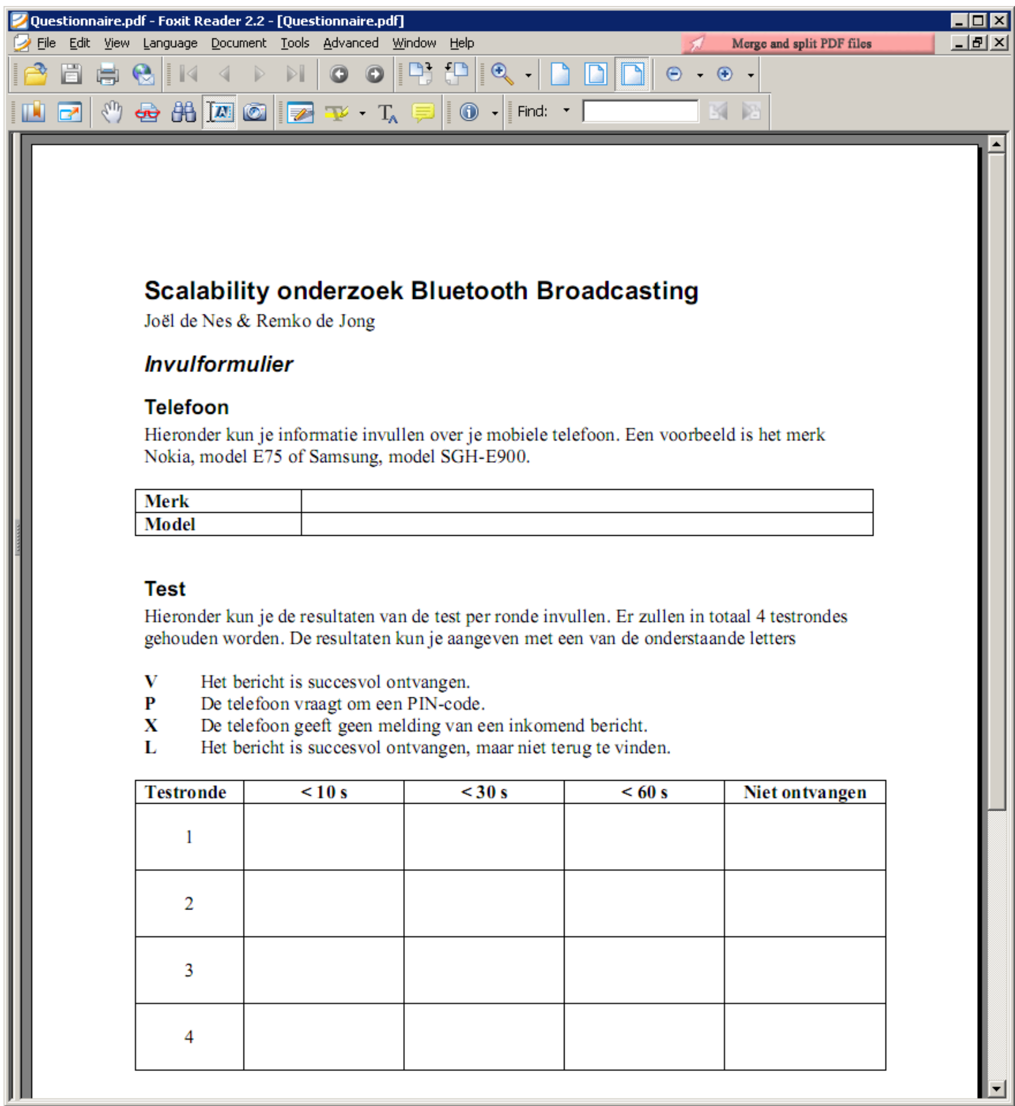
9.5 Content

Finally, we only experimented with simple text messages. Commercially various types of content are distributed: audio, video, pictures, business cards, etcetera. What are the possibilities for this system to distribute other types of content? What are other opportunities besides distributing Nestor announcements? Enabling students to download PowerPoint sheets during a course? And what is the impact on performance?

Of course these are just a few suggestions. When looking at this project from the perspective of the ever-continuing rise of pervasive computing a lot more research can be done.

Appendix A – Questionnaire

This appendix briefly discusses the questionnaire (cf. Figure A-1) that was used for the responsiveness test described in paragraph 6.3. The language used for the questionnaire is Dutch. First the participants were asked to fill in the brand and model of their mobile phone (*Merk* and *Model*). Their second task was to indicate the result of each test round with a letter, as explained on the form.



Scalability onderzoek Bluetooth Broadcasting

Joël de Nes & Remko de Jong

Invulformulier

Telefoon

Hieronder kun je informatie invullen over je mobiele telefoon. Een voorbeeld is het merk Nokia, model E75 of Samsung, model SGH-E900.

Merk	
Model	

Test

Hieronder kun je de resultaten van de test per ronde invullen. Er zullen in totaal 4 testrondes gehouden worden. De resultaten kun je aangeven met een van de onderstaande letters

V Het bericht is succesvol ontvangen.
P De telefoon vraagt om een PIN-code.
X De telefoon geeft geen melding van een inkomend bericht.
L Het bericht is succesvol ontvangen, maar niet terug te vinden.

Testronde	< 10 s	< 30 s	< 60 s	Niet ontvangen
1				
2				
3				
4				

Figure A-1: Questionnaire used for speed test

Table A-1 is meant to clarify the indicators used and their meaning. There were four test rounds and during each round the participants were asked to denote an indicator in one of the squares, depending on whether they received the message within 10, 30 or 60 seconds (*< 10 s*, *< 30 s*, *< 60 s*) or did not receive the message at all ('*Niet ontvangen*').

Indicator	Description
V	Message successfully received.
P	Device asks for PIN-code for pairing.
X	Device does not mention an incoming message.
L	Message successfully received, but cannot be retrieved on the device.

For a further description of the speed test and its setup see Chapter 6 – Experimental setup.

Appendix B – Questionnaire Results

For a description of the indicators used below (V, X, L and P) see Appendix A – Questionnaire.

Brand	Model																
Nokia	N73		V			V					V						X
Samsung	SGH-G600				X		V						X				X
Samsung	SGH-G600				X				X				X				X
Nokia	E71				X				X		V						X
Sony Ericsson	C902	V							X				X			V	
Nokia	N95	V							X				X				X
Samsung	SGH-G600				X				X				X				X
Samsung	SGH-G600	V						L					X				X
Nokia	N95				X				X				X				X
Nokia	6300				X		V					V					X
Nokia	6300				X			V					X				X
LG	KC550		V				V						X				X
Samsung	E730				X			L			L			L			
Samsung	SGH-U800			L					X	L						L	

Samsung	SGH-D600				X				X				X		V		
LG	Shine				X				X			V					X
Sony Ericsson	Cyber-Shot				X				X			V					X
Nokia	E75				X				X				X				X
Samsung	E900				X		P						X				X
Sony Ericsson	N800i				X				X		V						X
Nokia	E66		V						X				X				X
Nokia	6300		V						X			V					X
Sony Ericsson	W810i	V				V						V					X
Samsung	E900	P							X	P							X
Nokia	N73		V						X			L					X
Apple	iPhone				X				X				X				X
Nokia	6310				X			V		V					V		
Nokia	E71				X				X		V						X
Nokia	N70			V					X				X				X
Samsung	SGH-M150	V				V					V						X
Apple	iPhone				X				X				X				X
Nokia	6230i				X				X				X				X
Nokia	6120			V					X			V					X

Nokia	E66				X				X		V						X
Samsung	SGH-G600				X				X			L		L			
Nokia					X				X				X		V		
NTC	Touch HD				X				X				X		L		

Abbreviations used

AFH	Adaptive Frequency Hopping
BAN	Body Area Network
EDR	Enhanced Data Rate
HCI	Host Controller Interface
L2CAP	Logical Link Control and Adaptation Protocol
LGPL	Lesser General Public License
LMP	Link Manager Protocol
OS	Operating System
NFC	Near Field Communication
QoS	Quality of Service
RFCOMM	Cable replacement protocol (Radio Frequency COMMunication)
R*u*G	Rijks*universiteit* Groningen
SDP	Service Discovery Protocol

References

[1] Alasdair Pinkerton, Pervasive Computing. *POSTnote of the Parliamentary Office of Science and Technology*, 2006, nr. 263.

[2] Bluetooth, How it works, http://www.bluetooth.com/Bluetooth/Technology/Works.

[3] Bluetooth Technology Basics, http://www.bluetooth.com/Bluetooth/Technology/Basics.htm.

[4] André B. Bondi, Characteristics of Scalability and Their Impact on Performance. *Proceedings of the 2nd international workshop on Software and performance*, 2000, pp. 195-203.

[5] Bluetooth Proximity Marketing, http://www.bluetomorrow.com/content/section/392/513.

[6] D. Tsiantar, Getting on Board. *Time Magazine,* April 10th 2006, Bonus section A1-A4.

[7] ZoomBroadcast, http://zoombroadcast.com/index.htm.

[8] BlueGiga Proximity Marketing Statistics, http://www.bluegiga.com/solutiongroup?g=Proximity+marketing.

[9] Industry Statistics, http://www.punchkickinteractive.com/mobile/mobile-marketing-101/industry-statistics.php.

[10] Blue Media Server - Bluetooth Marketing Questions, http://www.bluemediaserver.com/bluetooth-marketing-questions.html.

[11] Project Oxygen, http://oxygen.lcs.mit.edu.

[12] Project Aura, http://www.cs.cmu.edu/~aura.

[13] J. P. Sousa, D. Garlan, From Computers Everywhere to Tasks Anywhere: The Aura Approach. Carnegie Mellon University, Pittsburgh, 2001, 17p.

[14] ZigBee Alliance, http://www.zigbee.org.

[15] How does ZigBee compare to Bluetooth, http://www.zigbee.org/About/FAQ/tabid/192/Default.aspx#18.

[16] How does ZigBee compare to other wireless Standards, http://www.zigbee.org/About/FAQ/tabid/192/Default.aspx#17.

[17] Near Field Communication Forum, http://www.nfc-forum.org/home.

[18] BlueCasting, http://www.bluecasting.com.

[19] Mobitouch^3, http://www.mobitouchcube.com.

[20] BlueSixty, http://www.bluesixty.nl.

[21] BLIPsystems, http://www.blipsystems.com.

[22] Bloo2 http://www.bloo2.com.

[23] BlueBlitz, http://www.blueblitz.com.

[24] BroadTooth, http://www.londondev.com/londondev/broadtooth.

[25] Alterwave, http://www.alterwave.com.

[26] S. Dalmolen, J. Hafkenscheid, B2Build. Rijks*universiteit* Groningen, Department of Computing Science, 2008, 23p.

[27] P. Murphy, E. Welsh, J.P. Frantz, Using Bluetooth for Short-Term Ad Hoc Connections Between Moving Vehicles - A Feasibility Study. *White Paper*, Rice University, Houston, Department of ECE, 2002, 5p.

[28] L. Yan, L. Zhong, K.J. Niraj, Energy Comparison and Optimization of Wireless Body-Area Network Technologies. *Proceedings of the ICST 2nd international conference on Body area networks*, 2007, Article no. 8.

[29] J. Eliasson, M. Lundberg, P. Lindgren, Time Synchronous Bluetooth Sensor Networks. *Consumer Communications and Networking Conference*, 2006, Volume I, pp. 336-340

[30] P. Nurmi, J. Kukkonen, E. Lagerspetz, J. Suomela, P. Floréen, BeTelGeuse – A Tool for Bluetooth Data Gathering. *Proceedings of the ICST 2nd international conference on Body area networks*, 2007, Article no. 21.

[31] M. Lipphardt, H. Hellbrück, D. Pfisterer, S. Ransom, S. Fischer, Practical Experiences on Mobile Inter-Body-Area-Networking. *Proceedings of the ICST 2nd international conference on Body area networks*, 2007, Article no. 7.

[32] C. Lombriser, N.B. Bharatula, D. Roggen, On-Body Activity Recognition in a Dynamic Sensor Network. *Proceedings of the ICST 2nd international conference on Body area networks*, 2007, Article no. 17.

[33] R. Kettimuthu, S. Muthukrishnan, Is Bluetooth suitable for large-scale sensor networks? *White Paper*, IBM, 2005, 7p.

[34] BlueCove Java Library for Bluetooth, http://www.bluecove.org.

[35] R. K. Morrow, Indoor Radio Propagation and Bluetooth Useful Range. McGraw-Hill TELECOM, 2002.