Impact of multiple inquirers on the Bluetooth discovery process

And its application to localization

Anne Franssens
July, 2010

Committee:
Dr. A. Wombacher (University of Twente)
Dr. M. Bargh (Novay)
Dr. M. Wegdam (Novay / University of Twente)

UNIVERSITY OF TWENTE.

Abstract

This thesis describes a research on the impact of multiple searching Bluetooth devices on the Bluetooth discovery protocol.

Bluetooth is a standard which describes energy efficient wireless communication. Because of its characteristics, most mobile devices incorporate Bluetooth as their means of wireless inter-device communication. In order for two devices to communicate using Bluetooth, they first need to set up a connection. A device can start a scan (inquiry scan) to discover available devices in the area. On a low level a device can be discovered without that device physically notifying its user. This principle is exploited to trace people as they move.

This research focusses on the discovery time when using multiple inquirers (searching devices). This discovery time influences the performance of a localization system. A series of experiments is set up to test this performance. A subset of these experiments are then used to create a model for modeling the discovery time for multiple inquirers. This model is based on the empirical data for one inquirer.

At least twenty devices can be discovered by one inquiry scanner. All devices that are within range can be discovered in an average of 5 seconds if more than 1 inquirer is used. The dutycycle, which is the ratio of scanning versus backoff time of the inquirer, can be set to 6 and 7..8 periods. This ensures a minimal discovery time and low yet distributed backoff time. The number of inquirers that leads to the fastest discovery is 7, the maximum of the experiment. Because of collisions, a competition effect among inquirers exists. Modeling the discovery time and the number of inquirers leads to a model with acceptable accuracy.

Contents

List of Figures

List of Tables

1 Introduction

Suppose there is a large, multi-story department store at which a lot of different items are sold. It might even include a restaurant. Having a system that could track customers as they move around the store would have several advantages;

- one can determine the route customers most often take
- one can determine which set of products are most popular by evaluating waiting-times
- one can have an informed push-offer-on-demand system, to give clients that appear to be in doubt (by detecting that they stand at an area for a long time) a coupon via Bluetooth, which is for example only valid for ten minutes.
- and consecutively the client behavior can be identified. For example: people that come to eat in the restaurant, which products are they most interested in?

These kinds of behavioral aspects of clients could be very important for a marketing-strategic business plan. It would be known exactly how to set up the departments in a store, which adverts to place in which sections, etc.

Creating such a system still requires a lot of research. Although localization itself exists for many years, it is still not possible to trace random persons. At least not without them having to wear a traceable device of some sort. This thesis describes a research that is done in the BlueWhere project at Novay (former Telematics Institute). This BlueWhere system uses Bluetooth to trace people that have Bluetooth enabled on their mobile devices. This eliminates the need for them to wear a special badge, and for them to be consciously aware of the tracing process. An advantage is that random people can be traced this way, because many people carry a Bluetooth enabled device (around 19%, section A.2).

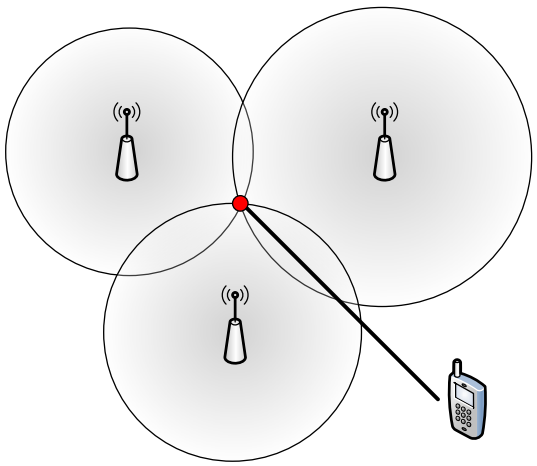


Figure 1.1: Trilateration to determine a location

1.1 Localization

For a long time people have had a wish for having some form of localization. Maps show that people have always had the wish to know their own position. Navigation based on stars, already used by the Egyptians, was the primary means of localization for many centuries.

Digital localization is a topic which acquired general attention in 1983 when U.S. president Reagan declassified the Global Positioning System (GPS). From that day on, consumers where able to use global satellite localization with a precision of about 100 meters due to Selective Availability (SA [36]). This military restriction was finally lifted in the year 2000, making civilian GPS precision of 10-15 meters possible. On a global level, it is therefore now possible to calculate ones own position. Despite outdoor localization, indoor localization is still a challenge.

There are two categories in which localization systems can be divided [6]:

- **Signal based**. This technique uses analysis of the properties of the wireless signal itself to calculate the position of the device. These signal properties include lateration, angulation and proximity detection [32]. GPS is processing based localization; it uses lateration and angulation techniques to determine positions by processing different parameters of the radio signals (e.g. RSS, angle of arrival, time of arrival, Δ time of arrival) [39].

  Example:

  Trilateration can be used to determine the position of a mobile phone. By estimating the distances of the cell phone from three different base stations, the approximate intersection of these distances indicate the position of the cell phone (see figure 1.1). This distance is measured by determining the difference in the

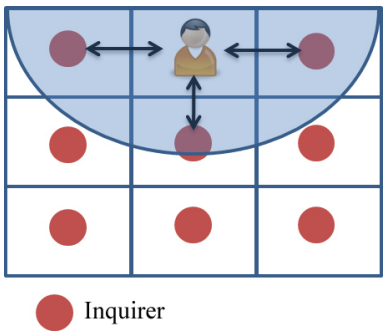


Figure 1.2: Fingerprint example

arrival-time of the signals of the mobile phone for each base station.

- **Fingerprint based**. Systems like this capture fingerprints of (some of the) known locations during an initial training phase. The network-characteristics of a device measured at each location are stored as a fingerprint. The measurement of a new device can be matched to these fingerprints. The best match determines the most likely location of the device. A fingerprint can for example consist of the IDs of the access points that are in range at a specific location.

  Example:

  Figure 1.2 shows a building with different rooms. Each room has its own inquirer. In the room at the top, a device is located. Because the range of this device is limited, it can only detect several access points from that room (within the drawn range). In the database it is known which fingerprint belongs to that room, i.e. which access points can be seen by the devices that are in that particular room. The combination of access points in the drawn range can see the device. By looking up this combination of access points in the database, it can be established that the user must be in that room.

  It is also possible for other 'processing based' radio information to be used. By combining for example the access points with RSSI values can yield more accurate results if done properly.

1.2 Bluetooth

The aim of the BlueWhere project is to enable localization using Bluetooth. There is one major advantage of choosing Bluetooth in favor of other types of technology; a lot of people already carry their localization tag A.2. While other technologies often require people to wear an identifiable badge or piece of equipment, Bluetooth is integrated in mobile devices many people already carry with them.

Other types of technology have also been explored in other research, such as GSM [25] and 802.11 (Wireless LAN) [11]. The latter unfortunately means that the traced object or person should carry an 802.11 enabled device. The first one might be of interest, but Bluetooth provides other significant advantages. GSM based localization is less accurate than Bluetooth localization, and does not work well in indoor environments. With Bluetooth it is possible to push software-objects to a device, enabling intelligent advertisement and potentially customer interaction [18]. Furthermore, it means operating in a freely available frequency-band, with hardware that can easily and cheaply be obtained at any computer store. By making these design choices, hopefully the project will lead to a relatively cheap and easy maintainable localization system, being able to trace all people that have Bluetooth enabled on their mobile devices.

BlueWhere relies heavily on the discovery (inquiry) process of Bluetooth. Because Bluetooth uses several frequencies to communicate, two devices must first synchronize the frequency on which they can do so [26]. The inquiry process is the way in which two devices can discover eachothers presence. It is this process that can be used to scan for people with Bluetooth enabled devices. Whereas the people carry the devices that have to be found (inquiry scanners), the localization system provides the devices that search for those devices (inquirers). When using more inquirers, inquiry scanners can be discovered faster up to a certain point. This principle is of great importance in this research. The next section discusses the problem statement.

1.3 Problem statement

An important issue in Bluetooth-based localization and tracing systems is the (adverse) impact of multiple Bluetooth devices on the performance of such systems. Several researches describe the theoretical impact multiple devices have, but only few provide measurements to validate these theoretical models. The primary objective of this research project is to study such impacts. To this end, experiments have been done (chapter 3) to measure the effect of multiple inquirers [1] and inquiry-scanners [2] on the performance of the discovery.

This research focuses on two major areas:

- the impact of multiple devices on the speed of discovery
- whether a model can be constructed to accurately model this performance

1.4 Research questions and contributions

The problem statement can be divided into two different areas, according to the areas already described.

Inquiry process (empirical studies)

As discussed in the problem statement, an important issue in Bluetooth localization systems is the impact of multiple Bluetooth devices on the performance

[1] For explanation of terminology, see section 2.2
[2] For explanation of terminology, see section 2.2

of the inquiry process, and thus on the localization capabilities. To this extent, experiments can be designed to asses the impact. The main research question is:

- How do multiple inquirers influence the discovery time for each inquiry scanner.

As every inquiry scanner needs to be discovered, taking a certain time t, each inquiry scanner has such a discovery time. The research question is accompanied by several sub-questions:

- **How many devices can be discovered**
  How many inquiry scanners can an inquirer detect in a reasonable amount of time. If this value turns out to be low, a localization system would not be feasible. It is therefore a basic test to asses the feasibility of the system in the first place.

- **Which dutycycle [3] is sufficient for continuous scanning**
  Chapter 2 describes the Bluetooth inquirer dutycycle in detail. In essence the research question is which ratio between scanning and backoff-time results in the lowers discovery time when aspiring to scan continuously.

- **What is the optimal number of inquirers**
  To reduce the discovery time more than one inquirer can be used. However, when using an infinite amount of inquirers, only collisions will occur. As a result, no inquiry scanners can be discovered. This means there is an optimal number of inquiry scanners.

- **Is there a competition effect among inquirers**
  The research on the previous research question closely related to a competition effect. This means that multiple inquirers try to find the same inquiry scanner, which will fail if there are too many inquirers. The gathered measurements that support the previous question can also be used to address this question.

- **Is there information in measurements related to distance**
  The measurements may reveal information related to distance. A well known candidate for this is the RSSI (Received Signal Strength Indication).

Modeling

The main research question for modeling the inquiry process is:

- **How accurate can the inquiry process be modeled using an empirical approach**
  Based on the answers to the research questions of the Bluetooth behavior, a practical model will be developed. Whether this model is valid and represents the actual measured results is also discussed. A subquestion is "can this model be made scalable so it can model more inquirers".

[3] For explanation of terminology, see chapter 2

Contributions

The contributions of this research are:

- **Discovery time with multiple inquirers**
  In this research the focus lies on the effect multiple inquirers have on the discovery time. Most other researches are only interested in the effect one inquirer has. If a variable number of devices is used, it is almost always the number of inquiry scanners.

- **Modeling based on practical data**
  As far as we know the existing literature does not provide with models based on practical observations. The models that we found rely solely on calculation based on the theoretical specification of Bluetooth, and have scalability problems. This research provides a model that requires calculation but is based on measurements instead of theory, and is scalable for multiple inquirers.

  As a side-effect, there are more contributions of this research:

- **Practical approach**
  Instead of predicting the outcomes by calculation, this research actually measures using multiple inquirers and inquiry scanners. This provides a large dataset on which a practical analysis can be based. Such a practical approach, including such a dataset, has not been described in papers before as far as we have found. If creating a model with such a practical approach is possible in the context of Bluetooth is a question which is answered in this research.

- **Observation window**
  Instead of using the start and end point of the dutycycle of an inquirer as a basis for measuring, an observation window is used (section 3.2). In short this means that this research looks at the discovery as an ongoing and uninterrupted process, which produces a stream of data with a large duration. The reason for this approach is that in a localization environment several inquirers are continuously scanning. Inquiry scanners can enter this scanning environment at any moment in time. When calculating the average discovery time for such an inquiry scanner, it is required to acknowledge that some of the inquirers may be in backoff mode. This paper is therefore not based on theoretical per-dutycycle behavior as other research does. The contribution is on one hand the very idea of the observation window, and on the other hand the way in which it is applied in this research.

- **Localization**
  By how many inquirers should each area or room be covered to provide a sufficiently low discovery time.

1.5 Outline

This thesis describes a part of the exploratory research of fundamental Bluetooth behavior. Chapter 2 gives a general introduction to Bluetooth, and in

particular the inquiry (discovery) process. Chapter 3 describes the experiments and conclusions regarding some fundamental parts of Bluetooth and its inquiry process. Chapter 4 describes the process towards a model for the performance of the inquiry process. Finally, a chapter is devoted to the tools and methods that have been used to measure and perform the research.

2 Introduction to Bluetooth

This chapter contains a general introduction to Bluetooth. Section 2.3 focusses on the inquiry process as a whole.

2.1 Introduction

The Bluetooth specification dates back to 1994. Jaap Haartsen, an electro technician employed at Ericsson Sweden, developed it in cooperation with Sven Mattisson. The name is based on the Danish word Blåtand, the tenth-century king of Denmark and Norway. The analogy with Bluetooth is in the uniting aspect. Whereas the king united the Scandinavian tribes into a single kingdom, Bluetooth unites different communication protocols in a universal standard. In 1998 the Bluetooth Special Interest Group (SIG) was founded, in which a lot of big companies took part.

Bluetooth was developed because there was a need for cheap radio communication among mobile phones and peripherals. Cables where thus being replaced by a short-range radio connection. Due to SIGs decision to make Bluetooth an open and royalty-free standard, it is still de facto standard for short-range wireless communication in WPAN (Wireless Personal Area Networking) situations.

Bluetooth operates in the 2.4GHz short-range radio frequency spectrum, which is a globally unlicensed frequency. In the available frequency band, 79 sub-frequencies are used to transmit data using Frequency-Hopping Spread Spectrum (FHSS). The modulation on to the carrier frequency is done by using Gaussian Frequency-Shift Keying (GFSK).

Bluetooth transmission power, and therefore its approximate range, is divided in so called power-range-classes (see table 2.1).

Class	Transmission power	Range
Class 1	100 mW (20dBm)	100m
Class 2	2.5 mW (4dBm)	10m
Class 3	1 mW (0dBm)	1m

Table 2.1: Bluetooth classes

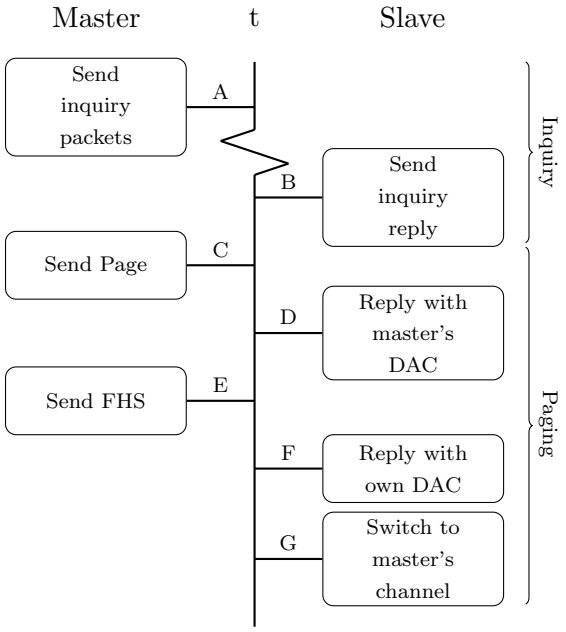


Figure 2.1: Inquiry and paging procedure

To enable communication among multiple devices from different vendors, not only a hardware or communication specification suffices. On the protocol level several standards must be specified to enable for example audio or data streams to be correctly interpreted by all devices. To tackle this problem, Bluetooth devices must be compatible with so called profiles. Popular profiles include for example A2DP for stereo audio, SIM for data from a mobile phone's SIM card and GOEP, the General Object Exchange Profile. If a profile is missing, the service the protocol provides can not be used. For example, the Apple iPhone 3G only supported the Hands-Free Profile and the Headset Profile [19]. As of a later release, more profiles such as A2DP have been added. Therefore, users can not use external GPS devices, and formerly could not share contacts or exchange files.

Bluetooth currently is used at version 2.0, supporting data rates of 3Mbit/s. Version 3 was announced on April 21 2009, supporting data rates of up to 24Mbit/s. Unlike previous versions, this version is based on WLAN (802.11n) making it incompatible with previous versions. In this research we will focus on Bluetooth version 1.2. Differences with other Bluetooth versions will, if relevant, be specified.

2.2 Data communication

A number of preset steps need to be performed in order to set up a connection between two devices. First of all, the other device needs to be discovered. This is done by the inquiry process which hops through a specified subset of all frequencies to find the devices that are discoverable (figure 2.1 node A and B). It retrieves their 48-bits unique MAC address and the internal clock-offset. It

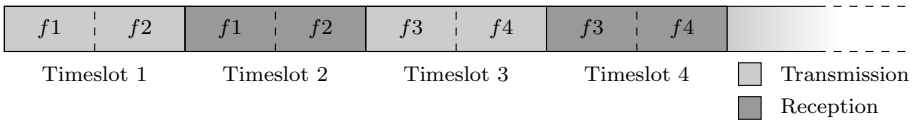


Figure 2.2: Inquiry process per timeslot of inquirer

is exactly this process which is used in this research in depth and described in the next section.

After this discovery has completed, a paging procedure is started to actually set up a connection (figure 2.1 node C until G). The master device pages the slave device, which in return sends a reply containing its Device Access Code (DAC) on the appropriate frequency selected by the page response hopping sequence. The slave will then switch to the master's channel parameters, by which a link is established and data can be exchanged. Most often this is done in the form of pairing. Pairs of devices negotiate a link key, a shared secret with which cryptographical authentication takes place. The stream of data may then be encrypted to prevent successful eavesdropping.

In order to communicate, Bluetooth uses a slow-hop frequency hopping spread spectrum scheme. This scheme consists of 79 frequency bands of 1MHz each, in the 2.4GHz range. In order to be incorporated into a Bluetooth *piconet* (a Bluetooth network), the device must be discovered in order to be able to exchange information to synchronize the hop sequence. Every piconet contains one master device, and up to seven active slaves. The master coordinates the transmissions of itself and its slaves by alternating in 625μs timeslots between master and slaves using time-division multiplexing.

2.3 Inquiry process

This research focuses on the behavior of Bluetooth devices during their inquiry process. As mentioned in the previous section, the inquiry process is designed to scan for other devices within range, and exchange the necessary information to set up an actual connection.

Bluetooth devices have two major states, *connection* and *standby*, and seven substates. Connection is used for communication whereas standby is the power-save mode in which no transmissions occur. The substates are used for joining as a slave in a piconet. The *page* substate is used by the master for adding slaves. For this *paging* procedure, the clock counter (28-bit, CLK) and the MAC address of the devices must be used. In the inquiry procedure this information is exchanged in order to set up a lasting connection. As the master sends its address and clock value, the slave can construct the correct hopping sequence of the piconet by that information. The master also provides the slave with a 3-bit identification number. This limits the number of slaves in a piconet to seven.

In order to exchange this kind of information between devices, a process must take place to actually find each other (discover). During the inquiry (discovery) process, the master enters the *inquiry* substate, whereas the slaves enter the *inquiry scan* substate. When the master of the piconet is in the

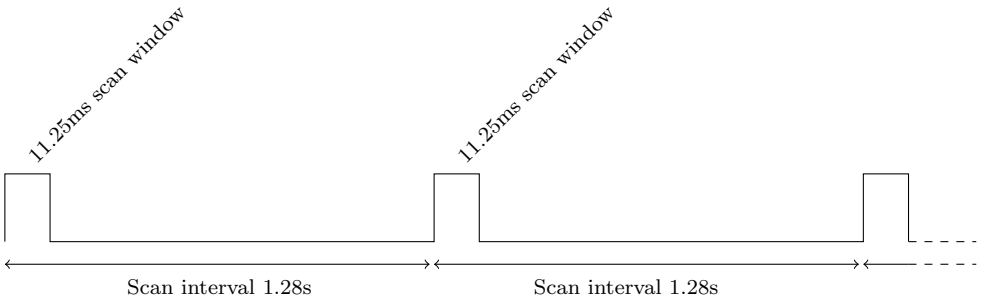


Figure 2.3: Scan windows per scan interval

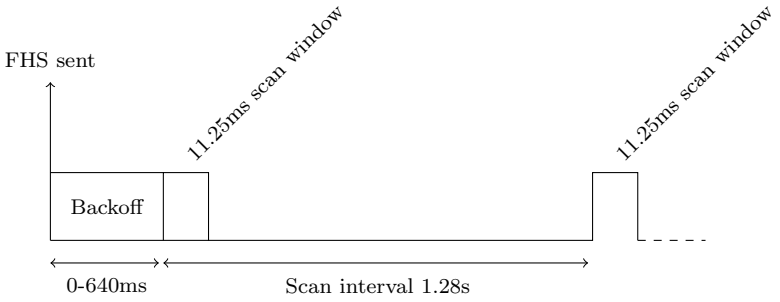


Figure 2.4: Inquiry scanner behavior after FHS reply

inquiry substate, all communication in the piconet is put on hold. Therefore it is crucial to keep the inquiry window short. Another reason for keeping the inquiry window short is that it scans/transmits via a large subset of the Bluetooth frequencies, thereby interfering with other piconets that may be within range. Therefore the inquiry procedure is optimized to find all devices in the lowest amount of time possible. The master transmits inquiry packets on different frequencies (figure 2.1 node A). The slave, called *inquiry scanner*, scans those frequencies at a slower rate, thus maximizing the probability of a correct reception.

An inquirer transmits two inquiry packets on two different frequencies during one regular transmission timeslot. 625 μs later, the inquirer listens on the same fequency (figure 2.2). The inquiry scanners, in scan mode, change the frequency on which they listen every 1.28 seconds. In those 1.28 seconds they scan for 11.25 ms only (figure 2.3). After receiving an inquiry packet, the inquiry scanner replies with an FHS (Frequency Hopping Synchronization) packet 625 μs later (figure 2.1 node B), and enters a backoff period between 0 and 1024 timeslots (0-640ms, figure 2.4). This FHS packet contains the device's address, its clock offset and a CRC code. Using this information a link can be established. This link is established by having the devices enter the paging substate, and go through the steps C until G of figure 2.1. This paging procedure is not important in this research, as the discovery is essentially complete at that time. We refer to [35] for more information on the paging procedure.

The inquirer uses *frequency trains* to determine on which frequency the inquiry packets are transmitted. There are two frequency trains, A and B. At the start of the inquiry process, A and B both contain half of the 32 frequencies that are used. The inquirer then selects either A or B, and starts transmitting and scanning (figure 2.2) using that particular frequency sequence. After 1.28 seconds one frequency from both trains is swapped, so each contain one frequency of each other. The inquiry process then continues as normal. After 2.56 seconds, the entire train is swapped so that A becomes B and B becomes A. This means that after 2.56 seconds, the other 16 frequencies are used, enabling the inquirer to find the remaining devices that were not discovered earlier.

For details of the frequencies that are used during the inquiry process, their order and more information, see [22].

3 Bluetooth inquiry performance

Performance on itself is, just like Quality of Service (QoS), a term which requires a definition of which criteria are actually considered. In order to judge the inquiry process objectively, parameters of the inquiry process need to be defined which can be measured, monitored or derived. This chapter continues with defining the different parameters of the inquiry process. It also provides a way of judging these parameters to fit in the frame of the research on localization as it is being done at Novay. After defining these parameters, their influence on the inquiry process is measured and their different effects have been described, each having their own section in this chapter. The contribution of this chapter will be a conclusion as to how the discovery times change when multiple inquirers and inquiry scanners are used. The research question that will be answered in this chapter is

> How do multiple inquirers influence the discovery time for each inquiry scanner

The experiments have been designed to answer this question accordingly.

3.1 Inquiry process parameters

The Bluetooth inquiry process, as described in the previous chapter, has many different variable parameters. These consist of user definable and tunable parameters that control the process. In this particular order, the parameters that are considered to be useful for this research have been described in the upcoming sections.

Dutycycle

The dutycycle is a parameter concerning the configuration of inquirers before the inquiry process. In short, the dutycycle is the ratio between active scanning and backoff-time. Both the inquirer and inquiry scanner use specific dutycycles for scanning in order to preserve power consumption. Although the dutycycle of the inquiry scanner can be changed, in common (mobile) device implementations the dutycycle is often set to the default standard. As for localization

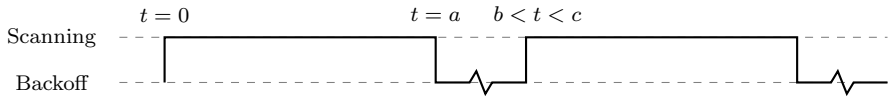


Figure 3.1: Dutycycle for (a,b,c) inquiry scan

the inquiry scanners can not be configured, the dutycycle is only accepted as a parameter for the inquirer.

The dutycycle can be specified by a series of three integers representing 1.28 second periods (see also figure 3.1):

$$(a, b, c)$$

where:

a. The periods the inquirer scans actively
b. The minimum number of periods of the dutycycle
c. The maximum number of periods of the dutycycle

This means that the backoff time is randomly chosen to end somewhere between b and c periods, leaving a minimum backoff period of $b - a$ and a maximum backoff period of $c - a$.

Example:

> *A dutycycle of (4,5,6) would scan for 4 periods, and backoff randomly somewhere between 1 and 2 periods.*

The influence of the dutycycle is twofold. First of all it determines the ratio of the amount of time that is actually spent in scanning mode. Secondly it determines the intervals at which the random backoff time is introduced. The average ratio of the time spent in scanning mode versus idle mode can be calculated using an equation:

$$\frac{a}{\left(\frac{(b-a)+(c-a)}{2}\right)} \tag{3.1}$$

which can be simplified in to:

$$\frac{2a}{b + c - 2a} \tag{3.2}$$

Example:

> *Suppose having a dutycycle of (4,5,6). This means the ratio of scanning versus idle time is*
>
> $$\frac{2 \cdot 4}{5 + 6 - 2 \cdot 4} = \frac{8}{3}$$
>
> *The interval at which the random backoff time is introduced is every* $a = 4$ *periods.*

When having a lot of different inquirers this possibly influences the results of the inquiry process, which requires investigation.

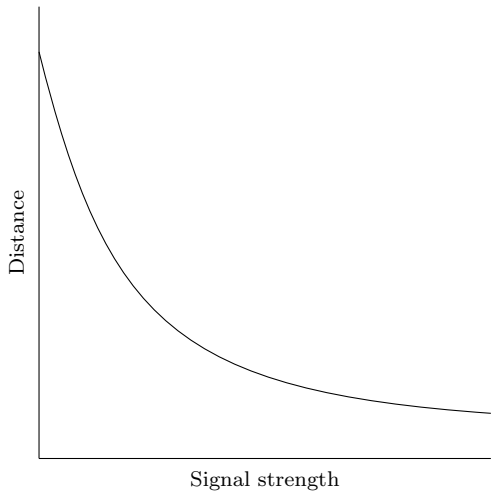


Figure 3.2: Distance versus perceived signal strength

Distance

Distance is also a parameter of the inquiry process. The inquirers and scanners can be placed at a certain distance d from each other, creating a larger or smaller gap to be bridged by the transmissions. One could also consider placing the inquirers at different distances, and all variations that can be derived from this principle. The influence of this distance on the inquiry process can thus be determined by doing repetitive measurements while varying d. It is already known that distance and perceived signal strength have a relation [17] (figure 3.2). The importance of the distance is therefore directly related to the localization problem. The effect of multipath fading and other signal distortion also have a relation with distance. This however can not easily be put in a graph as it relies heavily on environmental properties. A measurable influence of distance on the inquiry process therefore should exist. The results of this influence are discussed in section A.1.

Device type

Although Bluetooth devices are all built according to the same specification, there are several areas in which the manufacturer can make its own decisions. The specification also does not provide a detailed description of how a device should be built, it merely lists the requirements which it should fulfill. This means the manufacturer is free to choose how he actually implements for example

- the antenna
- the casing
- transmission power
- signal demodulation

These design choices influence the behavior of the system on all levels, including the inquiry process. Using different devices with, for example, different radio antennas, might reveal differences in the inquiry process. These however might be device related instead of particularly specific for the protocol itself. As the research goal is towards a practical localization system it is of interest how different devices behave. Nevertheless it needs to be kept in mind that obtained results can not simply be related to either the protocol or the device.

Power of transmission

The transmission power can be set in most devices. Manufacturers provide a required HCI [31] command set, which also provides support for getting and setting transmission power. As discussed in section 3.1 there is a relation between signal strength and distance. Changing the signal strength itself will therefore also be of influence. Multipath fading and other signal distorting influences also influence the perceived signal strength. Therefore there will also be a relation between these effects and the perceived signal strength.

Number of devices

The number of devices is also a parameter which can be changed. In a controlled environment this value can be explicitly selected. In field however, this value will be subject to constant change as people enter and leave transmission range. The effect of having multiple inquirers and inquiry scanners therefore is a parameter which is of great importance in this research.

When considering localization, the number of inquiry scanners represents the number of mobile devices that require localization. How many of those can be detected, and how fast, influences the performance of localization. The number of inquirers represent the number of access points that are used in a particular part of the localization area. Having more or less may influence how fast and complete the mobile devices can be detected.

As the number of inquiry frequencies is limited to 32, more inquirers will use up more of these frequencies. If the number of inquirers increases, the chance of collision between two or more will increase as well, thus influencing the results of the inquiry. Also, finding an inquiry scanner with multiple inquirers might be faster than with just one inquirer. These effects all determine the final shape of the inquiry process.

3.2 Experiment design

Designing the required experiments involves a few steps. First of all, the parameters that are subject to the test need to be determined. As it is too extensive to test every possible combination of parameters, a selection of those should be made. The aim of the selection is to provide results from which most research questions (section 1.4) can be answered.

After making this selection, the location and setup of the experiment needs to be chosen. To minimize anything interfering with the data that is collected, this needs to be carefully done. How the events that occur are actually recorded requires careful designing to minimize errors introduced during recording. When the recording of the data is completed, the data needs to

be processed in order to get the results which assist in answering the research questions. The recording process needs to be set up in such a way that processing the data can be done extensively, efficiently and correctly. Therefore, feedback after processing some initial measurements can assist in determining the final recording structure. The end of section 3.3 will discuss this.

Parameters

Of the parameters discussed in the previous section, a selection is made to use for the experiments. We have selected the parameters that are most likely to be of use in the area of localization. These are:

- dutycycle, because the amount of time spent scanning is important
- distance, because this is an important part of localization
- number of inquiring and inquiry scanning devices

The experiments for these parameters are split into two different groups. Before starting the actual experiment, the preferred dutycycle is established in a separate trial. When the dutycycle is selected, the experiments with distance and the number of devices can be conducted, using that particular dutycycle.

The remaining two parameters have not been selected as variable parameters in this research;

- **Device type** There are two reasons for not taking this variable into account. First of all a lot of different brands and types of devices are required in order to make a decent comparison. As the test should also be done with more than one device, it would be a logistical challenge to provide sufficient devices for the test. Secondly, it is very hard to draw conclusions from whatever results the experiments yield. The focus of the research lies on the real life behavior of the protocol during the inquiry process. The influence of different casings, antennas and transmission systems of all devices will make a comparison unreliable. The tests shall therefore be done using instances of one specific device type. Investigating if and how other devices behave differently is for future research.

- **Transmission power** It is generally possible to change the transmission power on Bluetooth devices. The HCI command set provides for the getting and setting of such a value. On most mobile devices, this setting is however not easily adjustable. Bluetooth dongles for PC-operation however are often equipped with a chip that supports this by easy configuration. It is a fair assumption that in the real world most devices will have the default transmission power setting. Usually this means maximum power for the Bluetooth class the device was built for. As the inquiry scanners in real world localization are mobile devices, their power setting is out of control of the system. During the experiments they will therefore not be changed. The power setting of the inquirers is also set to be fixed to limit the number of experiments.

As previously discussed the dutycycle experiment will be conducted in advance. Because choosing a particular dutycycle will influence the results of all

measurements, a selection of the distance/number of devices will be made and tested using different dutycycles. The results determine which dutycycle, if any in particular, is favorable. Table 3.1 contains the parameters with which the experiments have been done. Every combination of one value from each row makes for one individual experiment. Note that this adds up to 120 experiments. A single experiment will consist of the data of at least 100 executed dutycycles.

Dutycycles	(2,3,4) (2,4,5) (6,7,8) (20,21,22)
Distances	2
Number of inquirers	1,2,3,4,5,7
Number of inquiry scanners	1,5,10,15,20

Table 3.1: Parameters for dutycycle experiments

The rest of the experiments use the preferred dutycycle, in combination with the parameters of table 3.2.

Dutycycles	(a,b,c)
Distances	1,2,4,8,12
Number of inquirers	1,3,5,7
Number of inquiry scanners	1,5,10,15,20

Table 3.2: Parameters for other experiments

Note that this adds up to a total of 100 experiments, each consisting of the data of 250 executed dutycycles. This makes every experiment last for about 40 minutes (see section 3.3), leading to a total of $100 \cdot 40 = 4000$ minutes or 67 hours of collected data.

Choosing these specific distances, numbers of inquirers and numbers of inquiry scanners is done by consideration. The distances form a fair distribution on the maximum range of the used devices and their mobility. The number of inquirers is limited to seven for practical reasons. In a desired location-tracking system, having too many inquirers will increase the cost but probably not enhance the accuracy. As it is theoretically possible to track a 3d position with four devices, seven should at least suffice. As the used USB hubs have only seven output ports, seven is also a practical upper limit. When having one hub on the inquirer side of the setup, cable and hubs are left to spare for the inquiry scanner side, which is quantatively the most interesting side. On this inquiry scanner side, the maximum value is set to 20. There is no real theoretical reason for this particular upper bound. Practically speaking it is the largest amount of inquiry scanners the hardware can gracefully cope with when having three hubs.

Location and setup

Finding a suitable location to perform the experiments is not easy. The area needs to be as clear as possible from sources of interference, such as objects like furniture and interfering signals. At Novay, the souterrain basement (figure

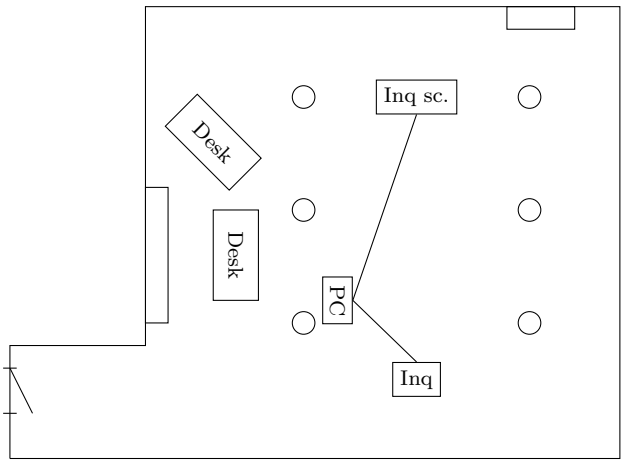


Figure 3.3: Novay basement

3.3) proved to be empty. Thick concrete walls and a minimal amount of objects occupying this space support a clean measuring environment. Signals can weakly penetrate this environment, like for example WLAN access points, but initial measurements however showed that Bluetooth signal penetration was very low. This most likely is the result of the larger signal strength WLAN uses by default. As the room is long, the experiments using maximum distance can still be performed by a clear line of sight between inquirers and inquiry scanners.

To perform the experiments, 4 USB hubs and two USB repeaters are used. Figure 3.4 shows the schematic design of this hardware setup. On one side, a USB hub is connected to the server. On the other side, two USB hubs are connected to the server, one of which is linked to yet another hub (figure 3.5). The capacity of each hub is 7 devices. On the inquirer side it is therefore possible to have up to 7 inquirers. On the other side it is possible to have $3 \cdot 7 - 1 = 20$ inquiry scanners. All devices that are put into the hubs are of the same type and have the same manufacturer: SiteCom, model no. CN-512 v2 001, with a Mavin Technology Inc (CSR 41B13) chipset.

Recording

Recording should be done in such a way that the recorded data can easily be processed to acquire the results. What actually is recorder is also subject of discussion, as the database should not get excessively large yet contain every bit of information that is useful in the research. Another aspect of recording is automation. As 67 hours of experiments have to be done, it is not desirable to have to do them by hand whilst changing the number of devices in the hubs every 40 minutes. A tool for recording and setting up the experiments automatically is therefore required.

A MySQL [21] database is selected for storing the data. This is mainly due to the easy setup, the easy integration of the database into programming languages and the performance of data retrieval. The automation is achieved by a tool, written in the programming language C. In short, the tool itself

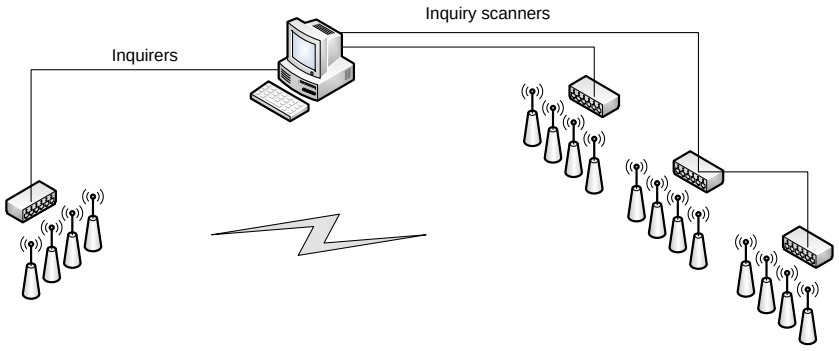


Figure 3.4: Experiment setup

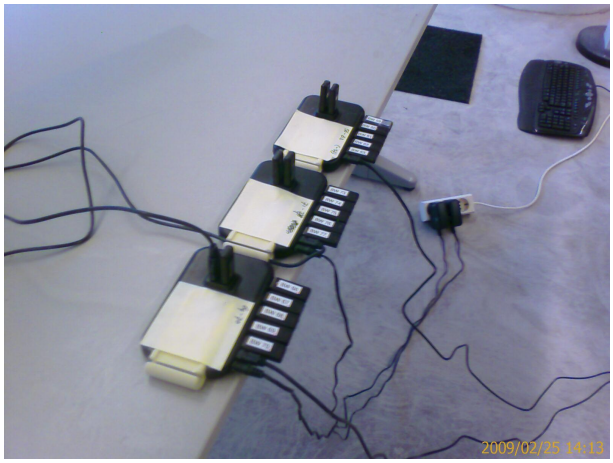


Figure 3.5: Novay basement with inquiry scanners

is a console program which is hacked against the linux BlueZ [1] Bluetooth protocol stack. It allows Bluetooth devices to be put in inquiry mode with all parameters set correctly. It also makes it possible to shut down devices completely, to ensure that they do not create noise on the Bluetooth frequency band. The tool captures every FHS reply packet, and allows for its content to be stored directly to the MySQL database. We refer to section A.3 for an elaborate overview of this tool. A Linux shell script is used to invoke the tool, using different commandline parameters which control the parameters of the experiment that is performed.

The recorded data is placed into a single database-table named "measurements". Figure 3.6 shows how the different tables that are used interact. Every table serves another purpose:

- **Experiments** For every experiment, this table records the parameters of the experiment, and a time of start and end.

- **Dutycycles** For every dutycycle of each inquirer in the experiment, its

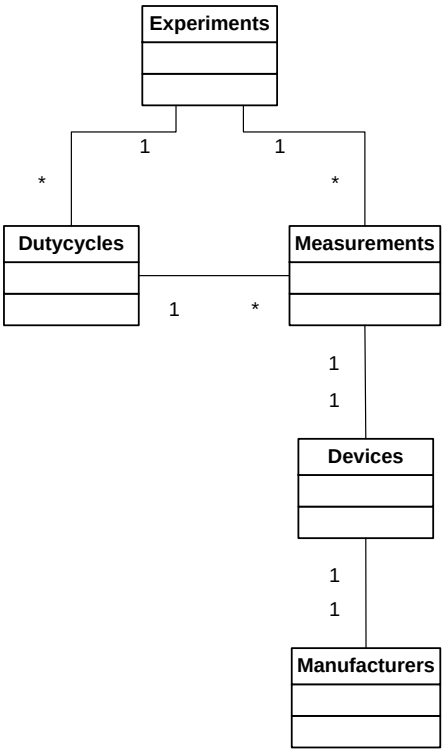


Figure 3.6: MySQL table structure

time of start and end is recorded. This way it will be possible to calculate at which time relatively to the dutycycle the event occurred.

- **Measurements** For every FHS that is returned by an inquiry scanner, the data package is stored in this table. This includes the RSSI value, and exact timing information timestamped by the server when the FHS arrives.

- **Devices** This table contains a list of all devices that are used in the test. It serves as a DNS for translation the text labels which have been attached to the devices, to their MAC addresses. If an unknown device is discovered during a test, it is added to this list with an empty label.

- **Manufacturers** This contains a list of Bluetooth manufacturers and the first three bytes of the MAC addresses they have been assigned to equip their products with. This way it is possible to determine the manufacturer for each device. Although this does not serve a real purpose in this research, it is nevertheless interesting when monitoring real mobile devices.

Processing results

Because the data is collected into a database, well designed database queries can make processing the results a lot easier. It needs to be kept in mind that processing the results should ideally be done in a short amount of time. Queries that take several minutes to complete make the analysis slow, and tweaking the process itself tedious. As the 'measurement'-table will contain a few million entries, this definitely requires consideration.

To make the results of the different experiments easily accessible and understandable, a series of web pages have been developed. These web pages list all experiments, and provide automatically generated graphs and statistical information for each experiment. Dynamically combining results from different experiments however is not possible, and should be done by hand or programming. Section A.3 shows the features and design of this web service, and presents the overview of all experiments. Using the framework built for that service, other information can be extracted more easily from the database and presented to the user accordingly.

Dutycycles and Observation Windows

There is one particular thing about the interpretation of the word *dutycycle* during the processing of the results that requires explanation. When several inquirers are involved in an experiment, each inquirer has its own dutycycle. As the backoff period is random for each inquirer, dutycycles itself are not synchronized. It would therefore make no sense to analyze the data of the different inquirers together, on a per dutycycle basis. In the field, when localizing and discovering new devices, a measurement starts at a certain moment. At that moment, it can not be predicted in which time and phase of the dutycycle the device currently is. A global "dutycycle", henceforth called "observation window", can therefore be introduced to counter that problem. Figure 3.7 shows how this can be achieved. As all inquirers have their own dutycycle, an observation window is introduced covering both the active and non active phase of the dutycycles. This observation window has no backoff time, as they are automatically included. Basically it comes down to congregating the measurements of all inquirers and analyzing them in predetermined intervals which are then called *observation windows*. Because all backoff times of the inquirers can be seen as random after some time, the global view will be valid. In terms of practical approach, this simulates the real world in the following way; devices enter and leave the vicinity of the monitored area. The inquirers are inquiring, all in different stages of their respective dutycycles. One of the questions is: how fast will they discover. To derive the distributions of such events, the data will be analyzed starting at some point in time, taking all incoming measurements from that time on, into account. This is what figure 3.7 represents.

3.3 Effect of dutycycles

Figure 3.8 shows four of the 24 graphs of the conducted experiments. The graphs show the average time that is needed for 3 inquirers to discover the various number of inquiry scanners within each ordinary dutycycle. Note that the data is aggregated over the three inquirers, and that the x-axis contains

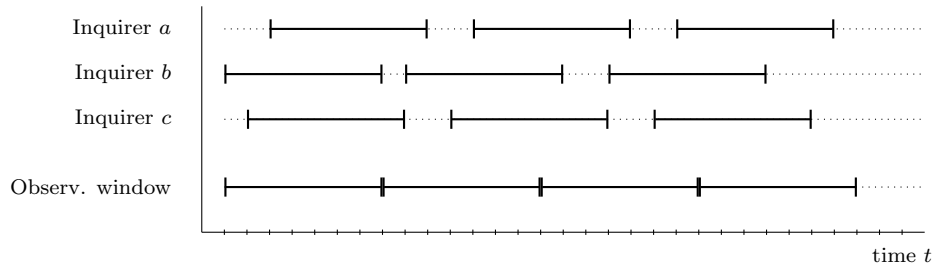


Figure 3.7: Congregation of dutycycles to "observation window"

$c+1$ seconds. The aggregation is done by mapping all dutycycles in one graph and derive the average. Because the data from the experiments also contains the start of all dutycycles for each inquirer, the timing for each inquirer can be very accurately shown.

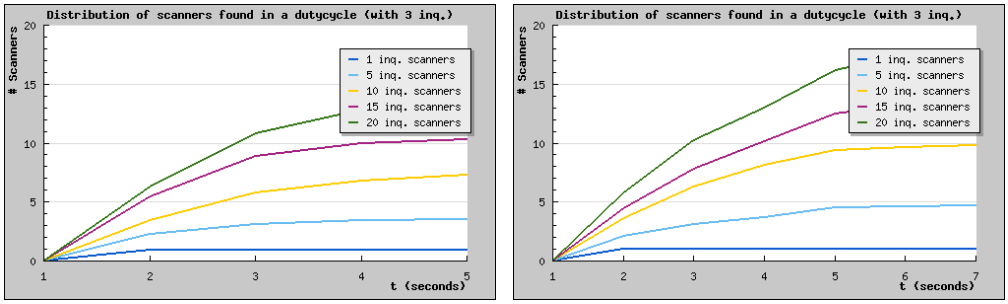
Example:

> *The green line in figure 3.8c represents an experiment with 20 inquiry scanners. As can be seen in the title of the graph, 3 inquirers were simultaneously used at that experiment. After 2 seconds in the observation window, on average 5 devices were discovered. After 5 seconds in the observation window, on average 16.5 devices were found.*

For determining the final dutycycle there are three criteria. On one hand the dutycycle must be short in order to maximize the amount of randomness created by the backoff period. On the other hand the dutycycle must be long enough to allow a maximum amount of discovery before the inquirer enters the backoff period. Also the dutycycle should have a small idle time to maximize efficiency. The dutycycle (2,3,4), for example, has an active scanning part of 2 periods, followed by on average 1.5 idle periods. This means that efficiency is low. The graphs show that there is little difference in the amount of devices that are discovered at a particular time t for different dutycycles. One exception is figure 3.8a, which discovers significantly less devices at $t = 5$. This dutycycle implies that only $2 \cdot 1.28 = 2.56$ seconds are actually spent in scanning mode. This means that it is possible that not all frequency trains have been covered by the inquirers, which results in a loss of overall discovery.

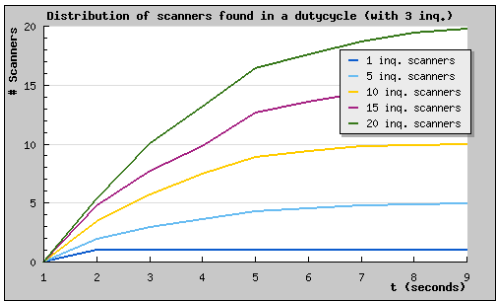
Not having large deviations in the measurements is an informal measure for the quality of the measurements. The time spent in scanning mode should theoretically not determine the number of devices that are found within a certain time t. The graphs clearly show that this is indeed the case.

Figure 3.9 shows figure 3.8c but expressed in percentages of discovered devices. At $t = 7$ an average of over 95% of all devices is discovered. This makes the dutycycle of (6,7,8), thus $6 \cdot 1.28 = 7.68s$, liable as a good dutycycle for the other experiments. It is relatively short, to allow the backoff periods to occur frequently. It is not too short in the sense that an average of 20% idle time is introduced. Furthermore, a discovery percentage of over 95% is considered to be a sufficient amount.

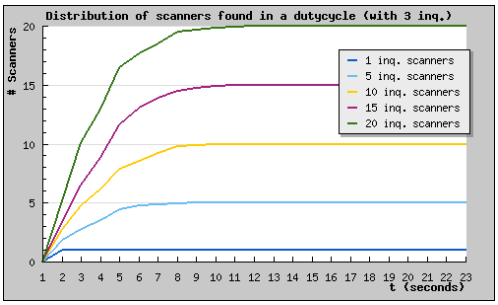


(a) Dutycycle of (2,3,4)

(b) Dutycycle of (4,5,6)



(c) Dutycycle of (6,7,8)



(d) Dutycycle of (20,21,22)

Figure 3.8: Time to discovery of inq. scanners, for different dutycycles

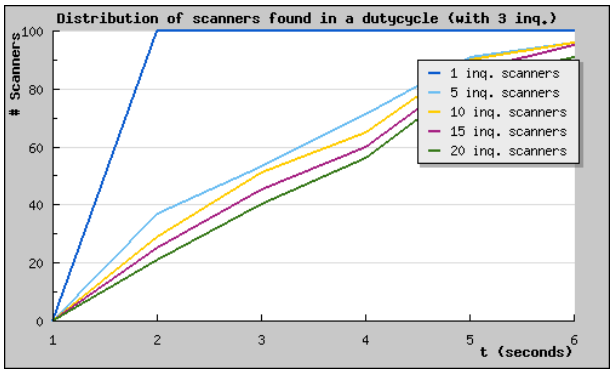


Figure 3.9: Figure 3.8c expressed in percentages

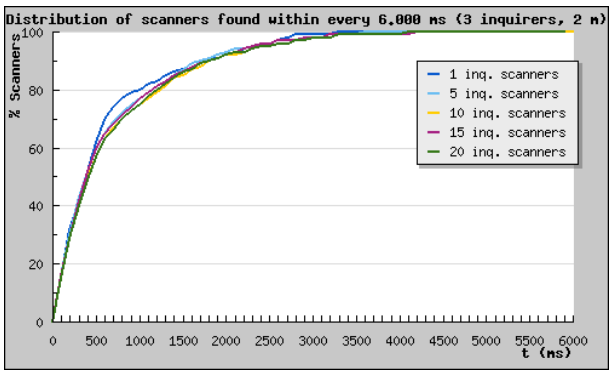
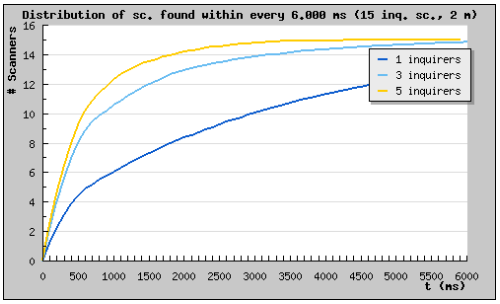
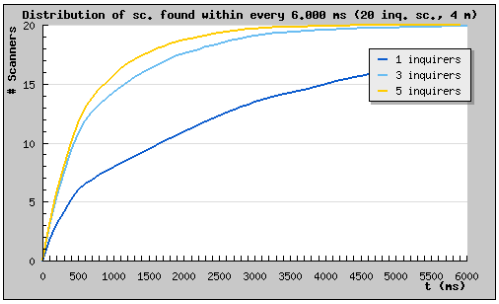


Figure 3.10: Figure 3.9 interpreted using a 6000ms observation window



(a) 15 inq. scanners at 2m



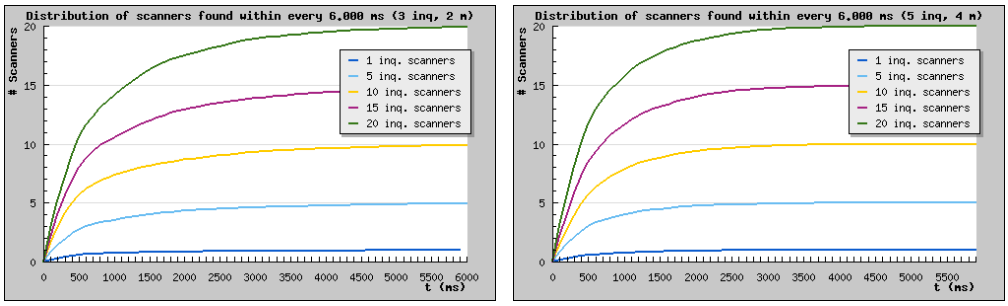
(b) 20 inq. scanners at 4m

Figure 3.11: Time to discovery of inq. scanners, for different number of inquirers

Although the (6,7,8) dutycycle is selected based on clear information and graphs, determining the dutycycle for which to carry out the experiments is not that easy. This is due to the different nature in which the dutycycles are interpreted in the other experiments (section 3.2), referred to as *observation window*. To verify the selected dutycycle, figure 3.9 is converted into an observation window interpretation (figure 3.10). An observation window of 6000ms is used instead of the individual dutycycles of the inquirers. It can be seen that the regular discovery percentage at 6s is exceeded. In fact, close to 100% is already achieved at the $t = 3500$ms mark. Besides showing other interesting behavior (see section 3.4), this means that the (6,7,8) dutycycle in combination with the 6s observation window is appropriate for the situation. They will therefore be used in the remaining part of this research.

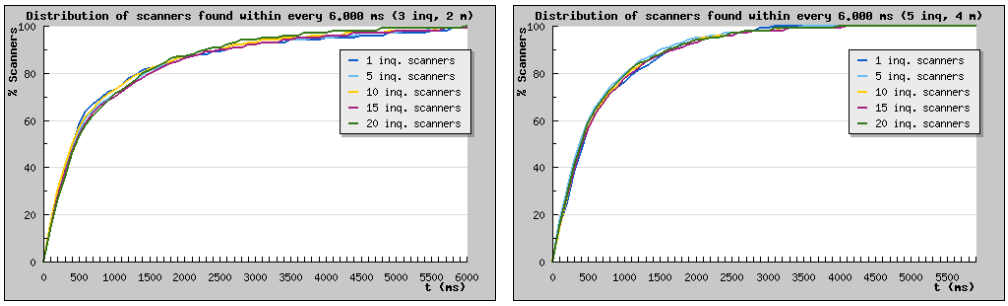
3.4 Effect of multiple devices

There are two ways in which multiple devices may have an impact. First the effect of multiple inquirers is discussed, followed by the effect of multiple inquiry scanners. This section contains the main contribution of this chapter.



(a) 3 inquirers at 2m (b) 5 inquirers at 4m

Figure 3.12: Time to discovery of different amounts of inq. scanners



(a) 3 inquirers at 2m (b) 5 inquirers at 4m

Figure 3.13: Time to discovery of different amounts of inq. scanners, in percentages

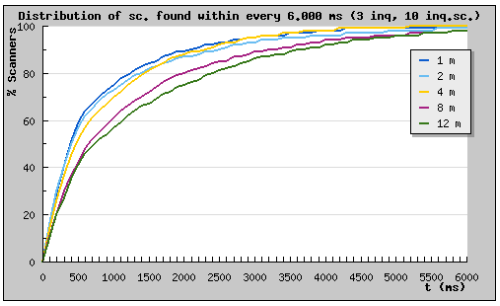
Effect of multiple inquirers

Figure 3.11 shows the general results of the experiments for a different number of inquirers. Note that from this moment on, observation windows are used instead of dutycycles.

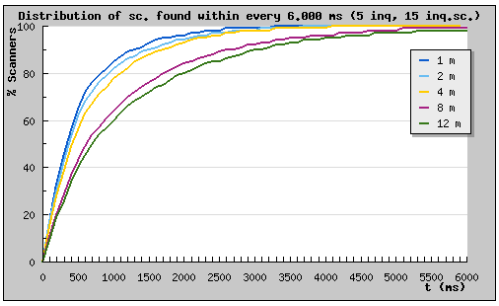
Example:

> *Observe the purple line of figure A.2a. This line shows the accumulated percentage of inquiry scanners that are discovered over time, by seven inquirers. For example, after* $t = 1000$*ms around 75% of the 15 inquiry scanners is discovered.*

The figures show that having one inquirer itself has a low discovery rate. From 3 inquirers on, the rates become stabilized. Note that there are no measurements for two inquirers, leaving the gap between the one and three inquirer graphs unfilled. It can therefore not be concluded if the two-inquirers graph would lean more towards the one-inquirer or the three-inquirer graph.



(a) 3 inquirers and 10 inquiry scanners



(b) 5 inquirers and 15 inquiry scanners

Figure 3.14: Time to discovery at different distances, in percentages

Effect of multiple inquiry scanners

Figure 3.12 shows the general results of the experiment with different amounts of inquiry scanners. Observe that having more inquiry scanners results in a better discovery of individual inquiry scanners. Figure 3.13 shows the same graph except the percentage of discovered devices is plotted. Introducing the observation window has introduced a behavior that we did not find in any literature. Apparently it does not matter how many inquiry scanners there are present, after a specific time t the same percentage of devices is discovered. All 20 of the generated graphs show exactly this same behavior. At first sight it may seem awkward, but section 3.6 will show that the measurements themselves are valid. Also the way in which the graphs are derived from the data has been thoroughly reviewed, and found to be correct.

We have not found this behavior documented in other research for version 1.2, therefore making this an interesting contribution to the general Bluetooth discovery research.

Statement:

> *With an arbitrary number of inquiring devices, between a certain time t and $t+\Delta t$, the same percentage of discoverable inquiry scanners within range is discovered. Provided the distance of the inquirers and the inquiry scanners is the same and the same conditions apply.*

This can be used for predicting the total number of scanners, given the number of found scanners in a period. The conclusion of the effect of multiple inquiry scanners is therefore easily drawn and can be summarized in the statement above.

3.5 Related work

The problem with comparing these results to other research lies in the fact that our approach is different. Most papers are interested in discovering multiple inquiry scanners using a single inquirer. We however are more interested in

having multiple inquirers, and the time in which the inquiry scanners are detected by at least one of these inquirers. When trying to improve the discovery protocol for communication, such an approach makes perfect sense. When the discovery is only used for aiding localization this is of no major importance. [26] and [10] are examples of papers that contain simulation results for one inquirer only. In [26] the same definition of a discovery is used; the time to receive and FHS packet from the inquiry scanner. After explaining the working principles of Bluetooth, they derive the timing information from the inquiry process. From this information, probabilistic information is extracted as to the hit and mis ratio of the frequency hopping of the FHS packets of the inquiry process. Although the paper is called "Bluetooth Discovery Time with Multiple Inquirers", they unfortunately conclude that an analytical model is not feasible, and "although the single inquiring device inquiry time has been well characterized, the effect of multiple inquirers is difficult to model and has not been considered". However, they do present a table containing the inquiry time with multiple inquirers, constructed by a simplified model in Matlab. This simulation models a single scanning node over a perfect channel in the presence of multiple inquirers. This results in a table with mean inquiry times for 1..5 inquirers. The value of $1.80s$ for 1 inquirer matches the results from this research, however the other values do not. Due to a different interpretation of discovery, the discovery times of the paper are significantly higher when compared to our discovery times, as they measure the time to have an inquiry scanner detected by each inquiry scanner instead of just one. Because we are interested in localization, we do not care which inquirer detects a device. Unfortunately this paper uses Bluetooth version 1.1. [10] is limited to a simulation for multiple inquiry scanners. After extensive analysis of the discovery protocol a simulator has been written using the original Bluetooth frequency train obtained from the specification. Although we have plenty of measurements of 1 inquirer with multiple inquiry scanners, the results can again not be compared. Whereas in the paper the inquirer has to find all inquiry scanners, we stop at the first inquiry scanner discovered. The results for 1 inquirer and 1 inquiry scanner however do match. In the paper they simulated an average discovery time of $1.5s$ in ideal circumstances, whereas we found it to be $1.8s$ in practice.

There is a reason why the values of this research can not be compared with Bluetooth version 1.1 reliably (as seen in [24] and [26]). In version 1.1, the inquiry scanner enters a backoff time after receiving the FHS. After that, a second discovery must take place by exactly the same inquirer, before the device is actually considered to be discovered. Due to this backoff time and same-inquirer requirement, the device is unable to be discovered by other inquirers for a considerable amount of time ([24]). In essence this leads to a larger mean discovery time when multiple inquirers are used on only one inquiry scanner. This is not the case in version 1.2, which requires only one discovery instead of two. In version 1.2, having multiple inquirers would not reduce the average discovery time, yet increase it. When comparing the values for only one inquiry scanner this effect is small, but when multiple inquiry scanners are used, the values can not be compared reliably.

In [24] a different approach is taken. An analysis is performed to obtain optimal parameters for the discovery phase thus proving that the default values are not optimal. A table is present showing the percentage of not-discovered devices, including the average discovery time. Although the paper uses Blue-

tooth 1.1 instead of 1.2, the value for one inquirer and one inquiry scanner can be used as a reference. For one inquirer and one inquiry scanner, 1.91 seconds is required on average to find a device. Although the distance between scanner and inquirer is not given, the average measurement we derive using version 1.2 with our observation window is 1.8 seconds.

In [16] a formal analysis made of the Bluetooth discovery protocol. This is done for version 1.1 as well as version 1.2. After an extensive discussion of the workings of the protocol, a probabilistic model is introduced. This model is based on the different aspects of the discovery phase. The extent in which the probabilistic part of the model checking is used is relatively small. Instead of using the absolute timing values, they are transformed to probabilities, so if a device is generally discovered in n timeslots with an even distribution, the chance of discovery in a timeslot is $\frac{1}{n}$. By using this approach, the influence of probabilistic modelling versus regular modelling is not large. The discovery procedure is broken down into a set of discrete time Markov chains (DTMCs). A setup for the probabilistic model checker PRISM is then introduced. The model is only used to calculate the values for one inquirer and one inquiry scanner. It is just this value that can be compared to our research. The simulation shows an even distribution of probabilities from $1.92s$ to $1.93s$ which therefore relates to our findings in the same way as the previously discussed papers do.

[9] is a paper on the the analysis of discovery and delay of Bluetooth devices. The aim is to provide an alternative backoff-time which lowers the overall discovery time yet preserves and respects the intentions of the back-off time. A bluetooth simulator has been written to simulate the behavior for different values for the number of backoff slots. Pracitcal measurements have also been done, but these are only mentioned to be in accordance with the values of the simulation. For the standard value of 512 backoff slots, the derived discovery time is $1.4s$ for one inquirer and one inquiry scanner. Although more inquiry scanners are used, their values can again not be compared. This paper concludes that the number of backoff slots proposed in the standard is too high. Equally good results can be obtained by reducing the number of backoff cycles to half its original amount, proposedly 200-300 ($\frac{512}{2}$).

In [7] the device discovery is approach from a scatternet perspective. The paper uses the fact that a scatternet is formed by the multi-hop wireless topology, requiring each pair of neighbouring devices to have a "symmetric" knowlegde of eachother. This means that if node u knows node v, then v knows u. This is indeed the case for Bluetooth, as correctly assumed. Using Bluetooth version 1.1, a simulation is created by using the VINT project network simulator. For one inquirer only, several inquiry scanners have been subject to this simulation, showing a result similar to our result in section 3.4. Although their result shows a slight deviation, the relative amount of discovered inquiry scanners within a time t is of a similar nature. Due to the difference in Bluetooth versions, the deviation can be explained. Whereas in Bluetooth version 1.1 a device needs to be confirmed after the first FHS packet by a second FHS packet, this is not the case in Bluetooth version 1.2. It can be observed that this difference in protocol version might well lead to a smaller deviation of relative discovery figures in version 1.2, thus confirming our finds. It can also be seen that the time it takes the inquirer to find these inquiry scanners in the paper is much higher than ours. In [7] 80% discovery is obtained after approximately

6.2s, whereas our experiments show that it is possible to achieve the same discovery in 4.6 seconds. As the simulation is always more optimistic than the experiment, this figure will show a larger deviation in practice. In short, the behavior shows characteristics that confirm the behavior we discovered, but due to a difference in Bluetooth versions can not be compared in detail.

3.6 Discussion

How reliable the obtained results are depends on several factors:

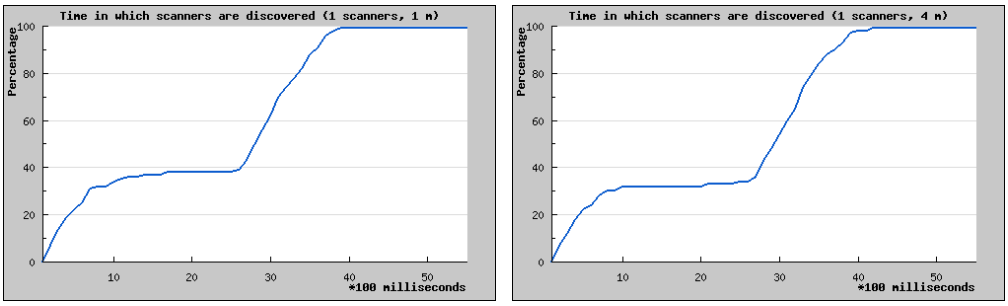
- **Environment** Changes in the environment, for example a moved object, can result in a difference in signal reception.
- **Climate** Changing humidity and temperature affect wireless signal transmissions.
- **Interference** Other radio-sources may interfere with the signals of the experiment.
- **Test setup** Moving test devices or placing them in an awkward position influences the way in which the transmissions perform.
- **Data collection** If for example measurement resolutions are very low, or if there are buffers that can get overflowed when not polling correctly the data collection can become unreliable.
- **Data Analysis** If the analysis is performed incorrectly data can be misinterpreted and result in conclusions that do not reflect the actual data.

The environment has not been subject to any physical change during the experiments. Furthermore, most of the influence of environment, climate and interference would have to be visible in the RSSI values of the measurements. It can be seen in the measurements that no significant changes in RSSI have occurred where they are not to be expected. This indicates that the influence of climate and interference was relatively low, although the experiments took several weeks to complete.

The test setup is easy, and does not show any signs for concern about the validity of the measurements it produces. Although the individual devices are relatively close to each other, this should based on the frequency hopping not be a problem. The vertical placement of the devices does require attention. During initial testing is was discovered that performance of the system was considerably worse if all devices were placed directly onto the floor. Probably an excess of relay scattering resulted in poor performance. Placing the devices about 1.5 meters of the ground proved to increase performance. Due to the concrete isolation of the room, signal penetration was low. In combination with the results of [20] this makes intereference an unlikely source of problems.

The data collection is partially discussed in section A.3, and should not influence the measurements too much. This is however a misleading assumption, as the subsection on FHS delays discusses later.

The data analysis is, on a high level, discussed in section A.3. Several of the data-analysis scripts have been externally reviewed by colleagues to ensure proper design and implementation. Especially the scripts that generated the



(a) Distance of 1m (b) Distance of 4m

Figure 3.15: FHS packets per dutycycle

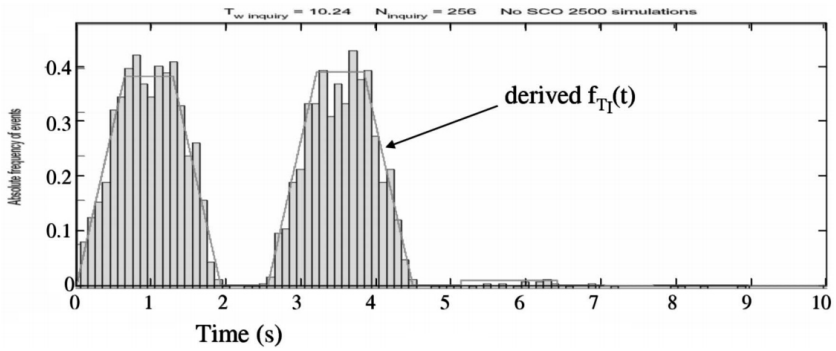


Figure 3.16: Simulated probability density for inquiry scan from [22]

graphs of section 3.4 have been subject to this review.

FHS packets

A more analytical approach to testing the reliability of the results can be performed using the actual low level measurements. During most of the analysis, only the first FHS packet of each inquiry scanner was used. As the discovery of a device depends upon the time at which this first packet is received, other packets received by this inquiry scanner in the same observation window are of no importance. When analyzing if the inquirers actually receive data in a way that can be matched to the theoretical approach of the Bluetooth protocol, these packets do have their use.

Figure 3.15 shows the total amount of FHS packets that arrive at a certain inquirer, averaged over all its original dutycycles. The graphs show a clear two-step phase in which the FHS packets have arrived. At time $t = 0$ the first batch of packets arrives. After some time this number stabilizes indicating that no new packets are arriving. Around $t = 2.5$ a new batch of packets starts arriving, behaving in the same way as the first batch.

A Bluetooth inquirer changes one frequency of the frequency train every 1.28 seconds (section 2.3). After 2.56 seconds the entire frequency chain is changed to contain all frequencies that were not present in the train of the first 2.56 seconds. The graphs show that the incoming FHS packet behavior is according to what should have been suspected when regarding the protocol. The first 1.28 seconds a lot of devices are found. The second 1.28 seconds only one frequency is changed in the train, thereby only changing by $\frac{1}{32}$. The influence of this on receiving FHS packets is small. After the 2.56 seconds, the entire frequency train changes allowing for almost all remaining inquiry scanners to be detected. The graph clearly shows a new batch of packets from inquiry scanners being received from that moment on. Figure 3.16 shows a simulation of the inquiry scan from [22]. This graph shows exactly the same behavior as the graph in figure 3.15.

FHS delays

For this approach the time between successive FHS packets from the point of view of the inquiry-scanners is taken. In other words, for every inquiry scanner, its Δ FHS is observed. Figure 3.17 (equals figure 2.4) shows the behavior of the inquiry scanner between successive FHS replies. After an FHS packet is sent, the inquiry scanner enters a maximum backoff period of 640ms. Immediately after this backoff time another scan window is opened. Therefore, another FHS can be discovered in the scan window immediately after the backoff, or $k \cdot 1.28s$ later. Using this theoretical insight, the reliability of the data can be determined.

Example:

- *If the backoff time was 0ms, the first scan window after the backoff time is immediately. The next one is then after 1.28 seconds.*
- *If the backoff time was 640ms, the first scan window after the backoff time is after 640ms. The next one is then after 1.92 seconds.*

If we consider all Δ FHS times lower than 1.2 seconds, this means that only the cases are considered in which the second FHS was received in the first scan window after the backoff time. These Δ FHS times should be lower than 640 + 11.25ms, because after that there is no scan window to receive them anymore. When the Δ exceeds this 651.25ms, this means that there was some form of a delay between the arrival of the FHS and its actual timestamp. This delay can happen when the Bluetooth chip reports the FHS later than it has actually arrived, or when there has been a delay between the chip reporting the FHS and its actual timestamping by the software of the measuring equipment. Figure 3.18 shows the Δ FHS times that are lower than 1.2 seconds. This figure shows that there are still packets timestamped after the 651.25ms mark, which ideally should not happen. This means that there is indeed a delay which occurs between FHS reception and its timestamp. Why this delay exactly occurs is very hard to determine. Although the tool that is used for timestamping the packets uses polling to acquire the messages from the Bluetooth devices, this does not introduce a significant delay by itself. In worst case there are only

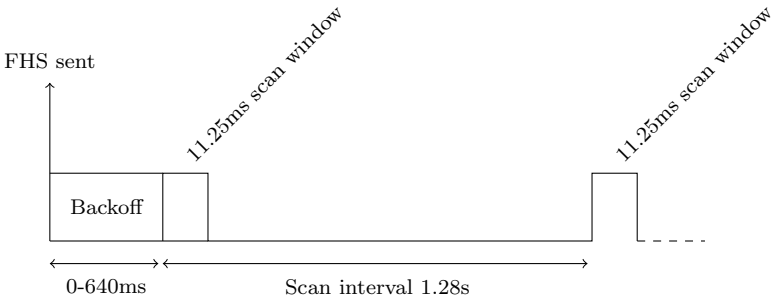


Figure 3.17: Inquiry scanner behavior after FHS reply

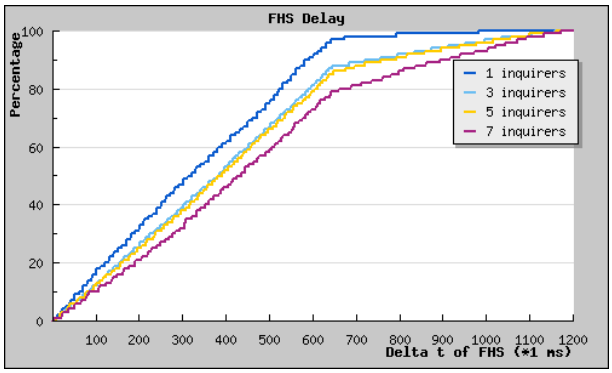


Figure 3.18: FHS Delay

7 inquirers that require polling. Therefore, the delay must be also present in either the used Bluetooth device, or the handling of Bluetooth packets by the USB hardware of the PC or the Operating System. In [28] it is suggested that messages may be delayed significantly when the device is busy:

> Unfortunately, the reply from the Bluetooth module may be significantly delayed if the Bluetooth module has received a data message via radio and is sending this message to the main processor ..., such that the communication channel ("Bluetooth-to-MCU channel") is blocked.

Although this will account for a small amount of errors, there still is the issue of multiple inquirers increasing the delay. This suggests that the USB hardware or the Operating System is most likely to be the major cause of this effect.

The experiment was carried out using three powered USB HUBs (section 3.2). The inquiry scanners located in two sequentially connected HUBs, and the inquirers located in one HUB. The HUB of the inquirers is connected to the PC by two long cables which include unpowered USB traffic repeaters. The PC itself contains two internal USB HUBs to which all devices are connected. The powered 7-port HUBs may consist of two cascaded 4-port HUBs [37]. In short, there is a long signal path from inquirer to the software. Every inquirer experiences different delays at 4 or 5 different stages. Each of those stages is not transparent as to which delays are introduced.

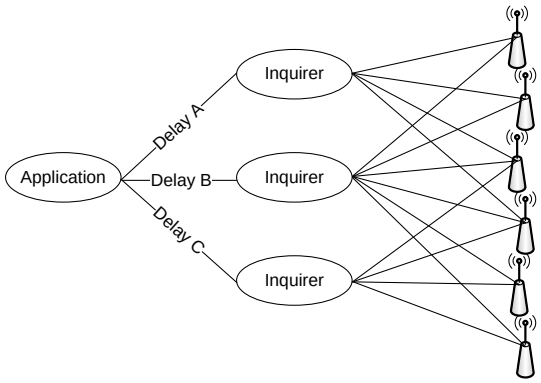


Figure 3.19: Delays between inquirers and application

In addition to that, the Host Controller (PC) directs traffic flow to the devices, which means that devices can only transfer data on the bus with an explicit request from the Host Controller. In USB 2.0 this is done by querying the connected devices, usually using a round-robin scheduling algorithm [38]. The PC, in this case, has a maximum of over thirty devices that require polling.

Where the delay exactly occurs is, as discussed, hard to determine. In chapter 4 a new experiment is suggested which removes the currently suspected causes of the delay.

The implication of these delays is that it is not possible to compare timing aspects of experiments with a different number of inquirers, as the number of inquirers show to influence the delay. Figure 3.19 shows the delays as they occur for each inquirer. Between the application and each inquirer a delay occurs. If it is assumed that the delay is on average a constant for every individual inquirer, the data for each inquirer is valid. If this is the case, delay {A,B,C} are not equal, and a time critical comparison between the results of these devices can not be made.

Therefore the results of this chapter are still valid:

- **Effect of dutycycles** This part does not compare different numbers of inquirers in a time critical way. It is based on how many devices are discovered in a certain dutycycle, which is not influenced by delay. Although the timestamp may be later, the actual dutycycle number in which the device was discovered still is valid.

- **Effect of multiple devices** The effect of multiple inquirers is based on observation windows instead of duty cycles. These rely on the timestamps of the FHS packets, and congregate the data of the entire experiment. If short and longer delays are experienced, they will even out to an average delay which is the same for experiments that have the same amount of inquirers. No conclusions based on data from experiments with a different number of inquirers have been made. The same holds for the effect of multiple inquiry scanners. The analysis is based on the observation windows, in which only the average delay is a factor. As the data is not compared to the data of experiments with a different number of inquirers in a time critical way, the conclusions are still valid.

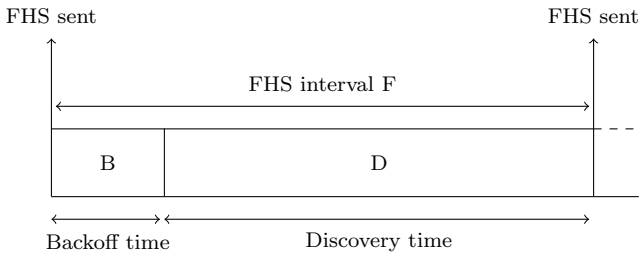


Figure 3.20: Inquiry scanner FHS Interval labelling

- **Effect of distance** The effect of distance is determined based on two different factors. The same experiments are repeated using different distances. As the delays of the timestamps can in that case be considered equal, the difference between the experiments at different distances is valid. For the second part, the research is based on the RSSI value of the measurements, not on the timing information of the measurements. Therefore delays do not influence the result.

As chapter 4 will require a detailed comparison of discovery times among experiments with different numbers of inquirers, the specific experiments for that chapter have been redone using a different approach (section 4.2.1).

3.7 Conclusion

The research question of this chapter has been answered by the statement made in section 3.4. To recapitulate, with an arbitrary number of inquiring devices, between a certain time t and $t+\Delta t$, the same percentage of discoverable inquiry scanners within range is discovered. Provided the distance of the inquirers and the inquiry scanners is the same and the same conditions apply. The number of inquiry scanners within range is therefore of no importance in this relation. It has to be noted that this conclusion is limited to up to 20 inquiry scanners, the maximum of the experiment.

4

Modeling discoveries

This chapter is aimed towards creating a simple but functional model of the actual inquiry process. Formal models based on a theoretical approach of Bluetooth already exist [22] [8]. A model based on collected measurements has however not been attempted to our knowledge. The measurements of the previous chapter have insufficient accuracy for this chapter. A new experiment has therefore been designed, and discussed in the next section.

To create a model, it must be determined what exactly will be modeled and how that should be done. In this chapter, two approaches for deriving a model are explored. The first model is based on the observation windows as introduced in the previous chapter. The second model is based on the FHS interval time. This is done to explore both possibilities, as well for their approach and results. The chapter ends with a discussion of both models and approaches.

This chapter will answer the research question:

> How accurate can the inquiry process be modeled using an empirical approach

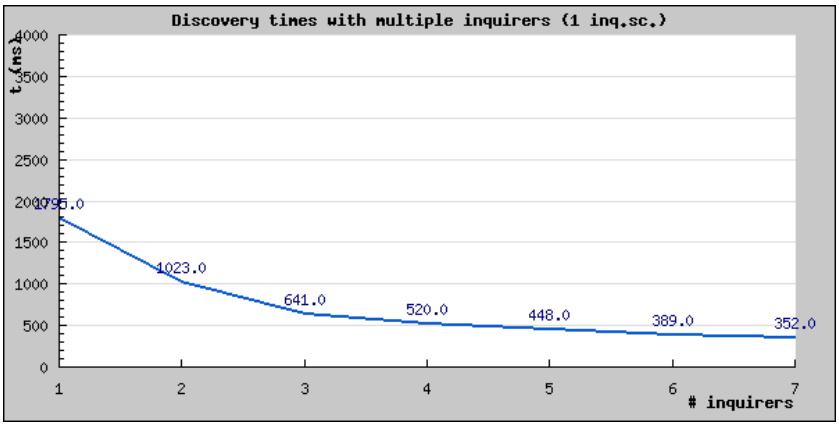


Figure 4.1: Inquirer dependent discovery times

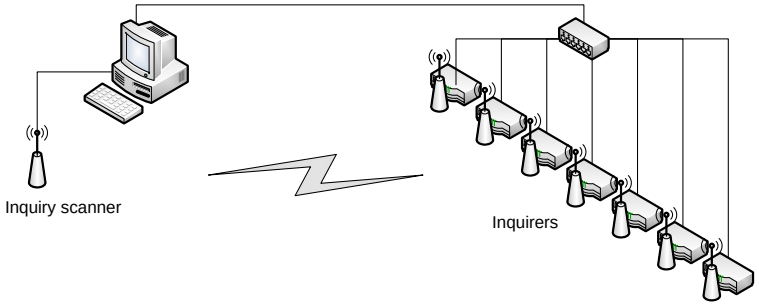


Figure 4.2: Experiment setup

and its subquestion of whether this model can be made scalable for multiple inquirers.

To this extent, two models using different approaches will be constructed, from which the best one is selected and evaluated.

4.1 Experiment design

As section 3.6 already described, the measurements of the previous dataset contain a delay. This delay is dependent on the number of inquirers. Therefore, an accurate comparison between discovery times of multiple inquiryscanners can not be made. In order to do this, a new experiment has been designed.

The delay of the data has been caused by either the USB bus or the PC on which the measurements were collected (section 3.6). The new experiment therefore has to improve these two conditions in order to provide more accurate measurements. In order to minimize traffic on the USB bus, each inquirer has its own device. In this case an ASUS WL500W [4] router. The routers have been fitted with a custom DD-WRT Linux operating system [14]. The routers are equipped with both a USB and LAN interface. The default DD-WRT Linux operating system does not natively support the Bluetooth library, and therefore requires external libraries.

Figure 4.2 shows the setup of the experiment. Seven routers with a Bluetooth module are connected to a LAN switch. The PC is connected to this switch, and has its own Bluetooth module. This way every device only has one Bluetooth module. The measuring tool (section A.3) has been rewritten and recompiled to support the DD-WRT routers. Because each router adds a timestamp to each FHS, the routers need to be synchronized. This is done by setting up a Network Time Protocol (NTP, [27] server on the PC, which synchronizes all connected routers every five minutes. The maximum skew is therefore reduced to < 5ms.

To be able to construct the model in this chapter, a subset of the previous experiments has to be redone. In this case the experiments containing 1 inquiry scanner and 1..7 inquirers have been redone. For every experiment 25 repetitions of 26 minutes of data have been stored. This leads to a total of around 4500 minutes of data. Figure 4.3 shows the delays of this new data.

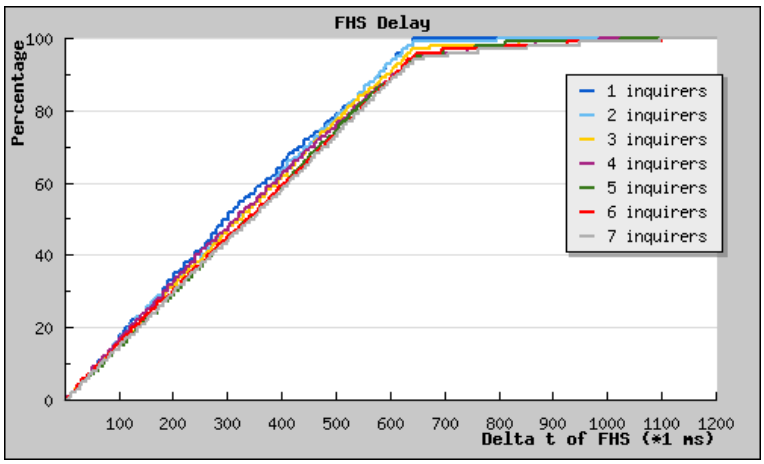


Figure 4.3: FHS Delays of new experiment

It can be seen that the delays have been reduced. The delays that still occur can be basically contributed to the same effects as the delays in the previous experiment, only to a much smaller intent. The quality of the data is sufficient for the purpose of modeling the discovery time. For quality assurance, it has been made sure that the data conforms to the standard shown in figure 3.15.

4.2 Model using observation windows

In this section the first model is derived. First an outline of the approach is presented, followed by the modeling and the result.

4.2.1 Approach

First it needs to be determined what is actually going to be modeled. In this case the dependency of the amount of inquirers on the discovery time is selected. Figure 4.1 shows this particular graph. On the vertical axis the time in milliseconds is plotted, against the number of inquirers on the horizontal axis. The graph shows the average time it takes for those inquirers to detect the available devices. To keep the modeling from being extraordinary difficult, only one inquiry scanner is used, positioned two meters from the inquirers.

The reaseon for modeling the average time to detect an inquiry scanner from any given point in time, is its usage in localization. When proximity based sensors are used, they are supposed to detect any (moving) inquiry scanner device as fast as possible. Modeling this behavior can assist in determining how to set up the localization system.

Modeling this process will be done by performing two consecutive steps:

1. Determining a basic graph
2. Fine-tuning the graph

The basic graph is calculated using probabilistic calculation by assuming that all inquirers behave in an independent way. If this would be true, adding

more inquirers will increase the speed of discovery. If an infinite number of inquirers would be used, there will always be an inquirer which has the same frequency as the inquiry scanner in the first timeslot. The time to discovery of the scanner would therefore approach zero. Using the measured discovery time from the experiment of one inquirer, the discovery times when having multiple inquirers can be calculated (section 4.2.2).

Because the inquirers are not independent (competition effect), these discovery times will be too optimistic. To correct for such influences they will have to be modeled themselves and combined with the independent model.

4.2.2 Modeling basic graph

In order the derive the basic graph the measurement of one inquirer is extrapolated in order to cover more inquirers. Using calculation the probabilities of discovering a device within t seconds can be derived. This can be intuitively illustrated by an example with dice.

Define:

$$\begin{aligned} p \equiv &\quad \text{Chance of throwing a six in a single turn} \equiv \frac{1}{6} \\ X \equiv &\quad \text{Turn in which 6 is thrown for the } \textbf{first} \text{ time by one of all dice} \\ X_1 \equiv &\quad X\text{, but for one die only} \\ n \equiv &\quad \text{Number of dice} \end{aligned}$$

For $n = 1$, so when having one die, the probability of throwing a six in the k^{th} turn is

$$P(X_1 = k) = (1-p)^{k-1} \cdot p \tag{4.1}$$

When having n dice the equation can be split into two parts; not throwing a six in the turns before k, and throwing a combination of up to n sixes in the k^{th} turn:

$$P(X_1 = k) = (1-p)^{n(k-1)} \cdot \sum_{m=1}^{n} \left(\binom{n}{m} p^m (1-p)^{n-m} \right) \tag{4.2}$$

The summation part of the equation can be rewritten, such that

$$\sum_{m=1}^{n} \binom{n}{m} p^m (1-p)^{n-m} \equiv \sum_{m=0}^{n} \left(\binom{n}{m} p^m (1-p)^{n-m} \right) - (1-p)^n \tag{4.3}$$

Using the binomial theorem of Newton the summation part of this rewrite can be reduced:

$$\begin{aligned} \sum_{m=0}^{n} \left(\binom{n}{m} p^m (1-p)^{n-m} \right) - (1-p)^n \equiv &\quad (p + (1-p))^n \\ &\quad -(1-p)^n \qquad (4.4) \\ \equiv &\quad 1 - (1-p)^n \qquad (4.5) \end{aligned}$$

This results in the following calculation for the probability of throwing a six in the k^{th} turn:

$$P(X_n = k) = (1-p)^{n(k-1)} \cdot (1-(1-p)^n) \quad (4.6)$$
$$= (1-p)^{n(k-1)} - (1-p)^{nk} \quad (4.7)$$

Equation 4.6 can be intuitively explained as follows: the probability of throwing a six for the first time in the k^{th} turn, equals the probability of not throwing it in the first $k-1$ turns, and one minus the probability that it is not discovered in this turn.

When using a more abstract notation the following equations state the desired probability. Note that the same deductive steps as above are used to simplify the equations.

$$P(X_n = k) = P(X_1 > k-1)^n \cdot \sum_{m=1}^{n} \left(\binom{n}{m} P(X_1 = k | X_1 > k-1)^m P(X_1 > k)^{n-m} \right) \quad (4.8)$$
$$= P(X_1 > k-1)^n - (1 - P(X_1 > k)^n) \quad (4.9)$$

In other words, the probability of throwing a six in the current (k^{th}) turn is the probability of not having thrown a six before, and one minus not throwing a six in the current turn.

Mapping this dice example to the actual inquirer model requires defining:

$$X \equiv \text{Millisecond in which the inquiry scanner is found by one of all inquirers} \quad (4.10)$$
$$X_1 \equiv X\text{, but for one inquirer only} \quad (4.11)$$
$$n \equiv \text{Number of inquirers} \quad (4.12)$$

The formula for deriving the probabilities per millisecond for multiple inquirers is equal to formula 4.9. Instead of evaluating the turn in which n dice are thrown, now the millisecond in which n inquirers are inquiring is modeled. The probability of finding the inquiry scanner with one of these inquirers, given that they are independent, is thereby calculated. The resulting model is shown in figure 4.4. The graph only goes down, as expected, until at some point it will approach zero. There is however a non-zero lower bound.

The lower bound of the discovery of one inquiry scanner can be derived from the Bluetooth specification. As an inquiry scanner is discovered, it generates a random number between 0..1023 and backs off for that amount of slots. This results in an average $\frac{1023 \cdot 625\mu s}{2} \approx 320$ms backoff time after a successful FHS transmission. This would mean that one scanner can be discovered no more than once every 320ms on average. The lower bound for the average discovery of one inquiry scanner is therefore 320ms. Figure 4.5 shows the result of this adaptation to the model.

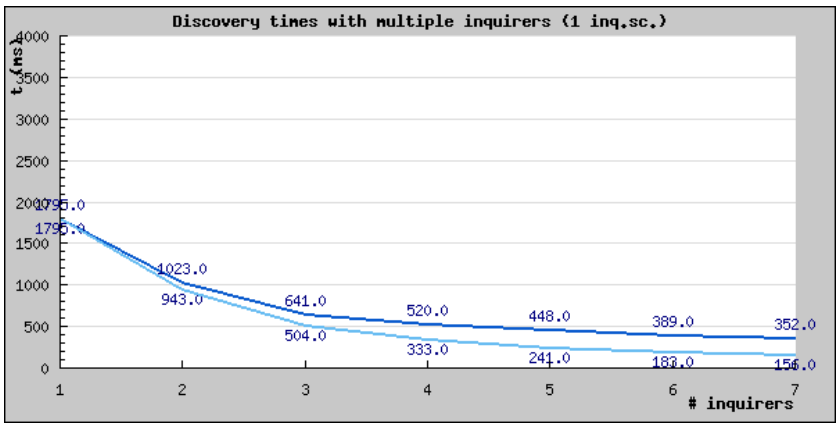


Figure 4.4: Inquirer dependent discovery times, including modeled version (lightblue)

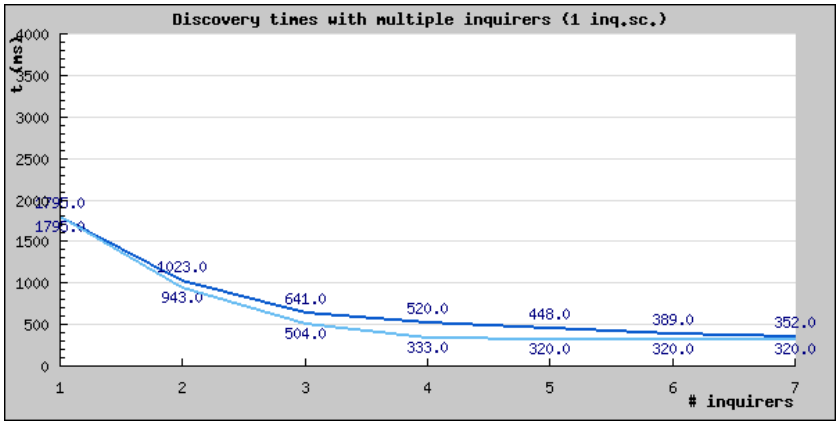


Figure 4.5: Inquirer dependent discovery times, including 320ms model lower bound (lightblue)

4.2.3 Fine-tuning graph

As the model from figure 4.4 is based on independent inquirers, the dependency needs to be added. In order to do that, it must first be established which factors contribute to dependency of inquirers. In other words, it must be established what the difference is between having one or multiple inquirers in a room.

Because the Bluetooth inquiry process uses frequency hopping, the only time that inquirers might influence each other is at exactly the same frequency. When two inquirers transmit on different frequencies there is no reason to expect them influencing each other on neither a protocol level nor on the radio transmission level. If two inquirers however do transmit on the same frequency, the result is a collision from which no data can be recovered by the inquiry scanner. Therefore the process will fail if two inquirers transmit on the frequency of the inquiry scanner at exactly the same time.

The effect of the independent model can be countered by applying this

knowledge in the form of a time correction. Whereas the independent model states that having more inquirers is always better, it can now be reasoned that there is an optimum present. Because the number of inquiry-frequencies is limited, having an infinite number of inquirers will result in permanent collisions on any frequency. This would mean that no device can be discovered at all, and will make the discovery time summit. Having a few inquirers would however be better than having one, because more frequencies can be inquired in less time, whilst the probability of having a collision remains low.

Collisions

Bluetooth uses 32 frequencies in order to locate inquiry scanners. A collision will occur if two or more inquirers transmit a package on the same frequency. How often this occurs can be calculated if the characteristics of the inquiry process are known. As creating an advanced model containing all subtle details is very hard, a simplified model will be created. This model assumes that all inquirers transmit their inquiry packets at exactly the same time, and are therefore synchronized.

First an optimistic solution to the model will be explained. This solution makes it easier to gain insight in how the actual created model works, which is explained afterwards.

The birthday paradox is a probability theory which describes the probability that in a set of randomly chosen people, two of them share the same birthday. In a group of 23 people for example, the probability is 50%. A well known application of this paradox is in breaking popular data encryption methods [13]. Using this paradox, the probability of two inquirers colliding on a frequency can be calculated. A problem with this theory is that the inquiry scanner scans every 1.28 seconds for only 11.25 milliseconds, but only on one frequency. This means that for a collision to happen, not only a collision must occur, but it must also happen on the correct channel. Otherwise the collision would have no impact on the discovery.

The solution for calculating the probabilities is calculating the probability tree. This tree will calculate the probability that a collision is the actual cause of not finding the inquiry scanner. An example tree is given in figure 4.6. In this tree, an inquiry process consisting of three slots and three inquirers is shown. Every line shows how many inquirers have the same frequency as the inquiry scanner, how many inquirers are left, and the probability that it happens. At the end of the last node the resulting probability is given, accompanied by a symbol which states the category of the value:

M = Match
NM = No match
C = No match, caused by a collision

An example will illustrate how this graph is set up.

Example:

The tree starts at the leftmost node, indicated by the start-label.

The line with "0/3" indicates that there are three inquirers, and zero of them have the same frequency as the inquiry scanner. Therefore no device will be found, and the same thing happens in the

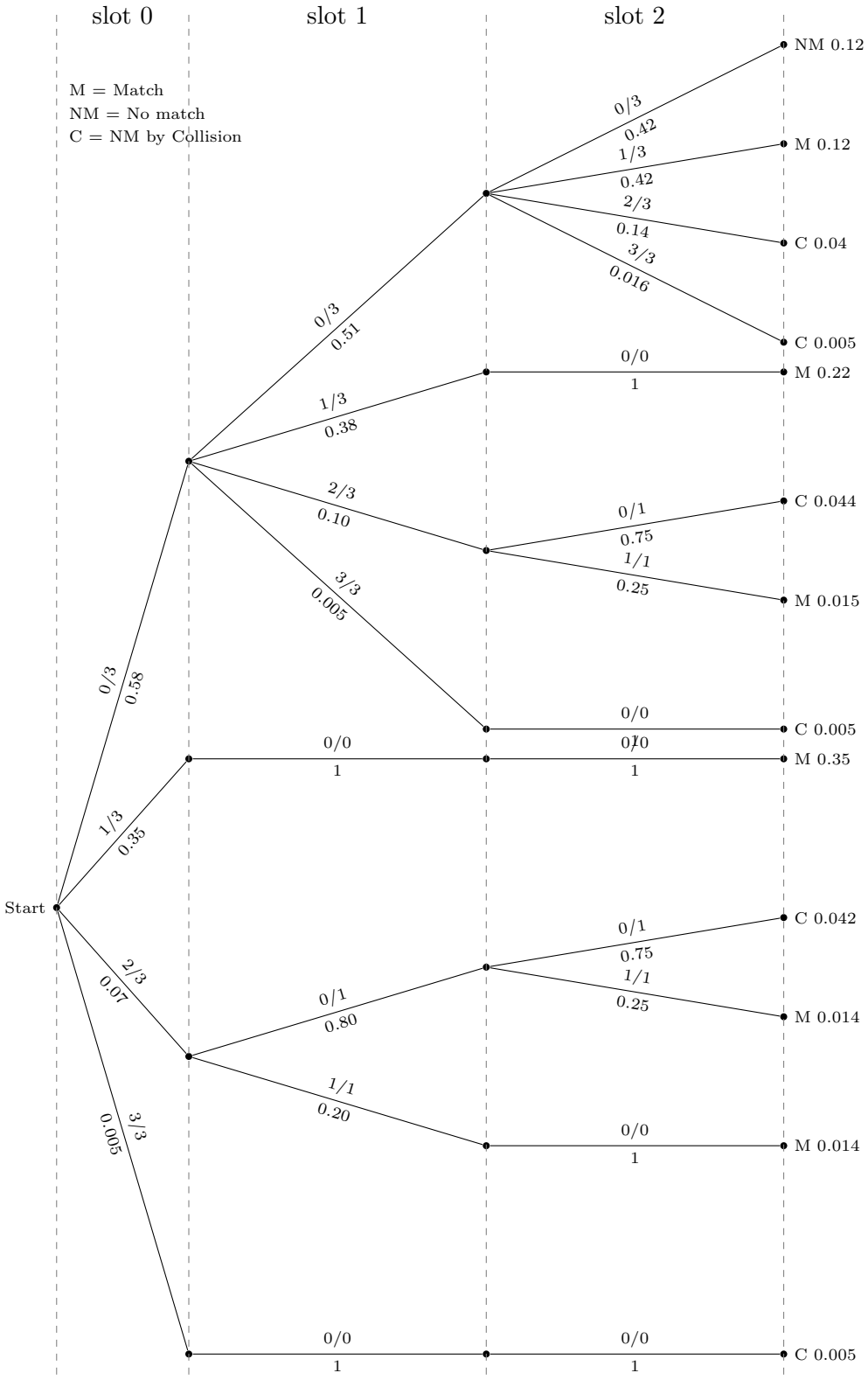


Figure 4.6: Collision probability tree

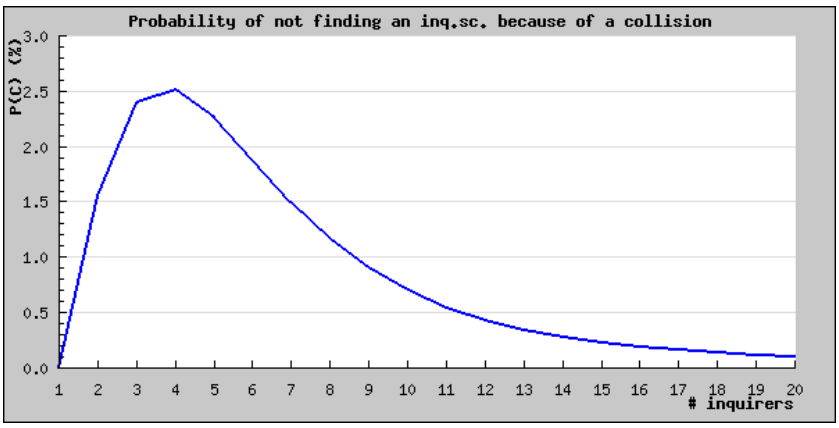


Figure 4.7: Collision probability, 20 inquirers

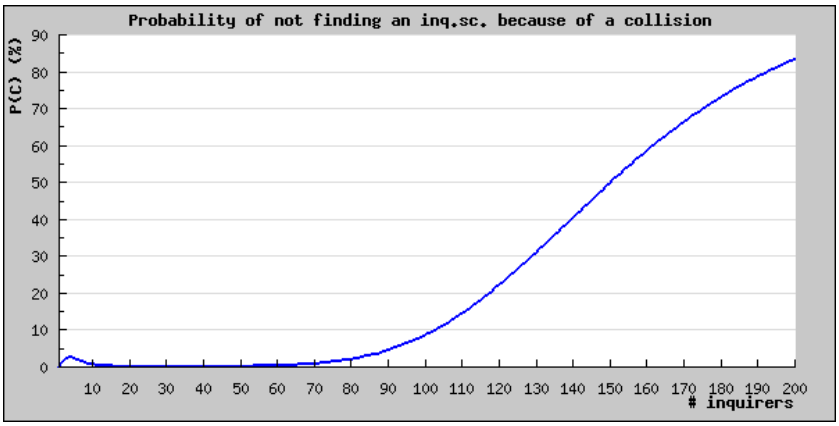


Figure 4.8: Collision probability, 200 inquirers

successive slot.

If we observe the second line from start, we see that one of the three inquirers has the correct frequency. The inquiry scanner is considered to be discovered.

The third line, labeled "2/3", has two inquirers which have the same frequency as the inquiry scanners. This means a collision has occurred. However, there is still one inquirer that has not visited that frequency yet, and still has a chance of making the discovery. In the second slot, the "0/1" line indicates that of the one remaining inquirer zero have found the inquiry scanner. If in slot three this inquirer again did not have the right frequency it results is a "C", because a collision (in slot 1) was the reason why the device was not discovered.

The probabilities can be calculated by applying a derived formula. Define:

$X \equiv$ Number of inquirers with the correct frequency
$N \equiv$ Number of inquirers
$F \equiv$ Number of frequencies (32)
$s \equiv$ Number of the slot, typically 0-15

$$P(X_s = k) = \binom{N}{k} \frac{1}{(F-s)^k} \left(\frac{F-s-1}{F-s}\right)^{N-k} \tag{4.13}$$

In short, it is the combination of ways in which k inquirers have the right frequency and the others have a different one. Note that as slots progress, less frequencies exist in the formula because they have already been used. In slot 0 there still are 32 possible frequencies from which to chose an appropriate value. In slot 1 only 31 different frequencies are available. Using this formula, the probabilities of each transition in the tree can be calculated. The tree is typically calculated for $s = 0..15$. This simulates one 11.25ms scan cycle of the inquiry scanner, which occurs every 1.28 seconds.

Using this tree, the probability of the match deficiency caused by collisions can be easily calculated using the probabilities of the leafs and their labels:

$$P(collision) = \sum P(C) \tag{4.14}$$

A tool has been written to calculate the values of this tree using a recursive algorithm. The resulting graph can be seen in figure 4.7 for a common amount of inquirers. The competition effect as suspected in the problem statement (section 1.3) is present and shown in figure 4.8, where more inquirers are included.

Having a discovery failing because of a collision will cause an average delay of 1.28 seconds before the next attempt can be made. The percentage of the 1.28 seconds average delay will therefore be added to the derived graph as this is its expected value. (figure 4.9).

4.2.4 Discussion

The model derived in this section is not a perfect model of the actual discovery times. However, when assessing figure 4.9 it is clear that the overall shape of the original graph is matched by the model. The deviations of the model form the actual data have been recorded in the table below.

Inquirers	Model deviation
1	0%
2	6%
3	20%
4	42%
5	28%
6	13%
7	4%

Although these deviations run up as high as 42%, the effort still produces a model that simulates the real behavior of the discovery. One of the reasons

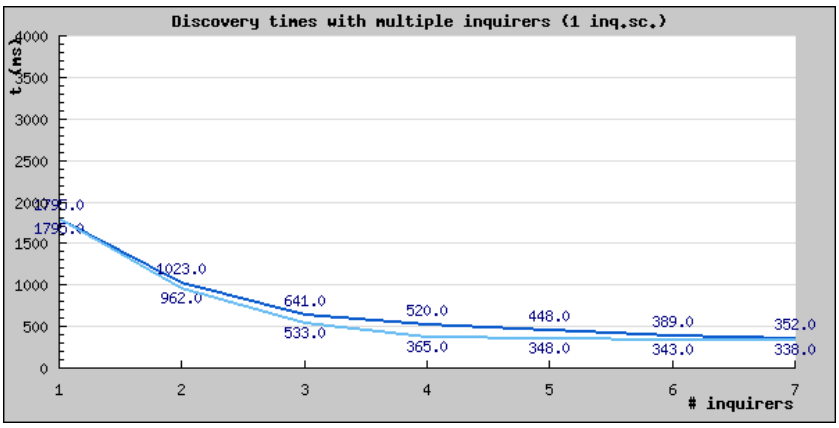


Figure 4.9: Inquirer dependent discovery times, including collision probabilities

why the model is always more optimistic than the actual recordings is because of the use of the observation windows. Whereas the data analysis incorporates the backoff time of the inquirers in its results for every inquirer, the model does not. The model is based on the information of 1 inquirer, including its observation window delays, but modeling based on that information might not be good enough for multiple inquirers.

In combination with the difficulties of other papers trying to model Bluetooth behavior, as [24] quotes:

> "It is very complex to study the probability distribution of this time in an analytical way"

We feel that our model approaches the real situation quite well. This means that although an analytical approach may be very difficult, deriving a good model is still possible. In retrospect, using a practical approach, it is not even that hard.

The research question states that the accuracy of the model is important. A discussion between both models can be found in section 4.5. The scalability of this model is very good. Both equation 4.9 and the equation for fine-tuning the graph 4.13 make the model scalable for as many inquirers as is required.

Comparing this model to other models described in literature is quite hard. First of all most models only attempt to model the behavior of one or two inquirers. Secondly none of the models use the observation window in the way we do. This research incorporates the backofftime of the inquirer into random positions of our observation window. All other research presented in section 4.4 provides models where the measurement is started at a specific instance, and ends on the end of the dutycycle, thus eliminating the backoff time and its influence completely.

For future work it is recommended to review the assumption that the observation window does not require any specific corrections for multiple inquirers. If this may be the case, and if it can be provided with evidence and a model of its own, the current model might be improved.

4.3 Model using FHS interval time

In this section the second model is presented, partially based on the approach of the first model. In line with the previous section, this section is divided into similar subsections. First an outline of the approach is presented, followed by the model and the results.

4.3.1 Approach

As with the previous model, the dependency of the amount of inquirers on the discovery time is selected. Instead of dividing the collection of measurements in observation windows, this time the FHS interval time is used. This measure is related to the response rate of an inquiry scanner to n inquirers. The response rate is used in indoor localization [6]. This model will determine how an inquiry scanner's response rate scales with respect to the number of inquirers (i.e. base stations used for indoor localization) at a given point. A higher response rate provides more information in a given time, which in turn increases the precision of such a localization system or equally enables faster tracking of (moving) inquiry scanners.

As shown again in figure 2.4, the FHS interval time (F) consists of two independent parts. After replying to an FHS the inquiry scanner enters a random backoff time of 0..1023 slots, or 0..640 milliseconds. This backoff time (B) is uniformly distributed according to the specification [9]. After the backoff is complete, the device is again available for detection. The duration from the end of the backoff period up until the next detection is the discovery time (D). Thus

$$F = B + D + \epsilon \tag{4.15}$$

where B is uniformly distributed over 0..640 milliseconds and the discovery time D has an unknown distribution. This distribution is not uniform due to the scan windows that occur on preset intervals. The ϵ corrects for noise in the equation. In this case that means noise in the distribution of B and D, and a possible small dependency of B and D. Hence forth this ϵ will be incorporated into D, as this is the distribution that is calculated using a convolution during the actual modeling process.

The model that is derived in this section is aimed at finding a model for the discovery time D, as this is the only unknown factor of the FHS relation. With that model, the overall FHS interval time can be modeled. The measured average infer FHS time is depicted in figure 4.10. Separating the FHS intervals F of the measurements into their respective B and D components is done by deconvoluting their two probability distributions. As they are independent, and F and B are known, this will result in a probability distribution for D. This probability distribution is then extrapolated to obtain the distributions for multiple inquirers, by the same method used in the previous model.

By placing the distribution of B back into the equation, the model of F for multiple inquirers is finished.

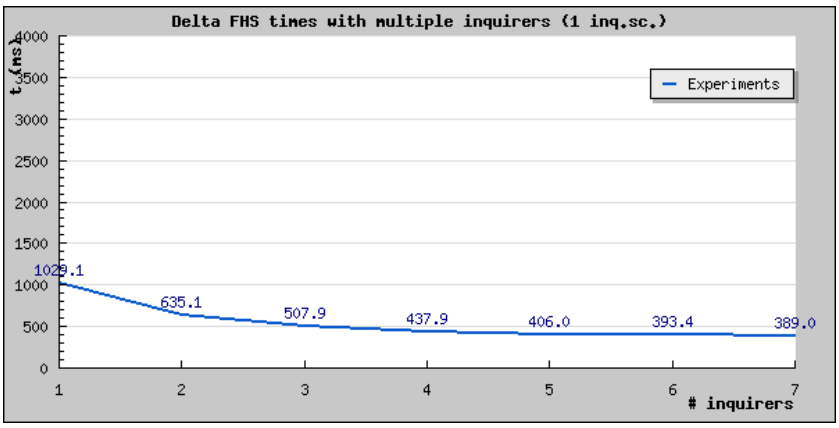


Figure 4.10: Average FHS interval times

4.3.2 Modeling

As both F and B are known distributions, D can be calculated using a regular discrete convolution.

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m]\, g[n-m] \tag{4.16}$$

In this particular case the convolution of B and D forming F can be seen as the following convolution:

$$F = (B * D)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} B[m]\, D[n-m] \tag{4.17}$$

Example:

Given the following probability distributions for B and D

$$\begin{array}{llll} B[0] & = 0.5 & D[0] & = 0.1 \\ B[1] & = 0.5 & D[1] & = 0.2 \\ & & D[2] & = 0.3 \\ & & D[3] & = 0.4 \end{array}$$

the discrete convolution of F is:

$$\begin{array}{lcll} F[0] & = & B[0] \cdot D[0] & = 0.05 \\ F[1] & = & (B[0] \cdot D[1]) + (B[1] \cdot D[0]) & = 0.15 \\ F[2] & = & (B[0] \cdot D[2]) + (B[1] \cdot D[1]) & = 0.25 \\ F[3] & = & (B[0] \cdot D[3]) + (B[1] \cdot D[2]) & = 0.35 \\ F[4] & = & B[1] \cdot D[3] & = 0.2 \end{array}$$

As the convolution is made up of two independent probability distributions the resulting distribution is also probabilistic. This means that the integral of

the resulting discrete probability density function (pdf) is always equal to 1, ergo

$$\int_{-\infty}^{\infty} F \equiv \sum_{m=-\infty}^{\infty} F[m] \equiv 1 \tag{4.18}$$

In order to calculate the values of D the probabilistic equation can be formulated:

$$P(F = k) \equiv \sum_{m=0}^{k_{max}} P(B = m) \cdot P(D = k - m | B = m) \tag{4.19}$$

If both D and B are independent, D does not rely on B and the second $B = m$ term can be removed, making the calculation of $P(D = k)$ possible. This will, if not entirely true, introduce a penalty in the form of ϵ.

After removing the $B = m$ dependency, the equation can be rewritten so it can be used to calculate $D = k$:

$$\begin{aligned} P(F = k) \equiv & \sum_{m=1}^{k_{max}} (P(B = m) \cdot P(D = k - m)) + (P(B = 0) \cdot P(D = k)) \\ P(D = k) \equiv & \frac{P(F = k) \cdot \sum_{m=1}^{k_{max}} P(B = m) \cdot P(D = k - m)}{P(B = 0)} \end{aligned} \tag{4.20}$$

As B has a uniform distribution, $P(B = \alpha), \alpha \in (0..k_{max})$ can be considered a constant $P(\beta)$, and the equation can be simplified to the following final equation

$$P(D = k) \equiv \frac{P(F = k) - \sum_{m=1}^{k_{max}} P(\beta) \cdot P(D = k - m)}{P(\beta)} \tag{4.21}$$

$$\equiv \frac{P(F = k) - P(\beta) \cdot \sum_{m=1}^{k_{max}} P(D = k - m)}{P(\beta)} \tag{4.22}$$

This formula calculates the distribution of D for 1 inquirer. Because of the possible noise ϵ, this is essentially $D + \epsilon$ where $\epsilon \approx 0$. If ϵ is large, this could result in a negative $P(D = k)$ if $P(F = k) < P(\beta) \cdot \sum_{m=1}^{k_{max}} P(D = k - m)$. A negative probability does not exist, therefore in order to compensate for this a negative $P(D = k)$ can be set to zero. By doing this, the integral of the resulting distribution can exceed the standard value of 1 for a probabilistic distribution. This effect requires compensating. During the discussion of the results (section 4.3.3) both cases will be explored.

To extrapolate this distribution to fit multiple independent inquirers, the formula derived in the previous model can be used (dice example, formula 4.9).

4.3.3 Results

The cumulative probability distribution of F, as can be derived from the measurements of the new experiment, is given in figure 4.11. Applying the convolution can be done in two ways; either by allowing only for positive values

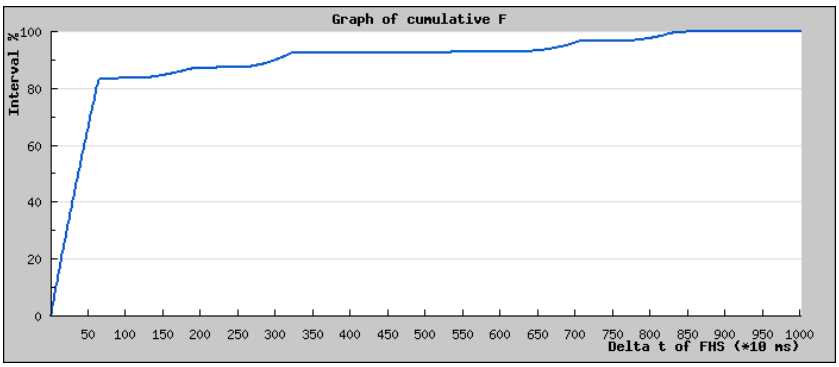


Figure 4.11: Cumulative probability density function of F

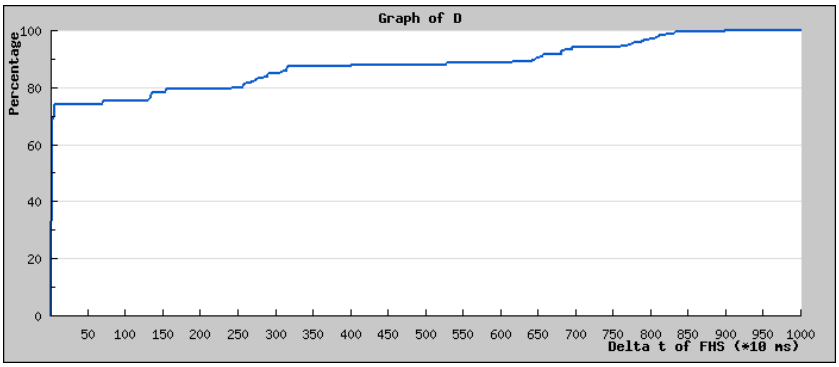


Figure 4.12: Cumulative scaled pdf of D^+

of D, or by also allowing negative values of D. In order to distinguish between these two approaches of D, they will be referred to as $D^+$and D^-. Figure 4.12 shows that graph of the only-positive Ds (D^+), figure 4.13 shows the graph of the possibly negative Ds (D^-).

The next two subsections describe the two models of F, $F^+$and F^-, which can be deduced from these models of D. As $F = B + D$, the average of F is composed of the average of D and the average of B. The averages of D can be calculated from the graphs of D. The average of B, due to its uniformity, is $\frac{640ms}{2} = 320ms$. By adding that 320 milliseconds to the average of D, the resulting F is calculated.

D^+

The graph of figure 4.12 shows that at the early phase of the actual discovery time, around 70 percent of the devices is already discovered. According to the graph of F around 80 percent of the devices should be discovered when the maximum backoff time of 640 milliseconds ends. Preventing negative behavior of $P(D = k)$ leads to an integral area of 1.34 of the overall distribution. In the graph there has been a correction applied to compensate for this behavior, but the fact remains that this value is much higher than it is supposed to be. In the equation (4.22) it can be seen that negative values of the term $P(D = k - m)$

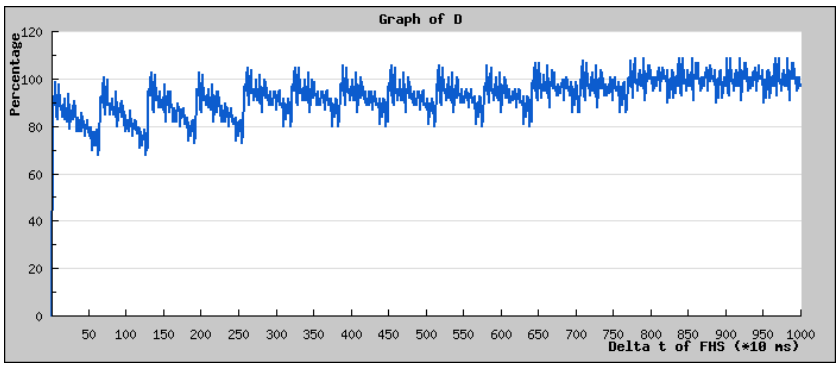


Figure 4.13: Cumulative pdf of D^-

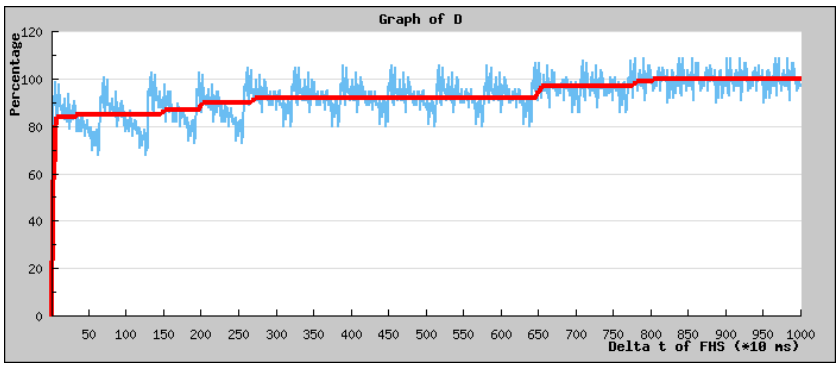


Figure 4.14: Cumulative pdf of D^-, with average

are subsequently used to calculate the value of a new $P(D = k)$ for a given k. Therefore negative values influence the way in which the equation behaves, and influences the other results. Setting these values to zero removes the auto-compensation that is inherent to the structure of the convolution, which in this case leads to an overshoot of 34 percent.

When using this graph of D^+, the resulting probabilities for multiple inquirers can be calculated, leading to the result of figure 4.15. Note that the scale of the x-axis is modified to exclude the large flat continuation of the graph for $\Delta t > 1000ms$. The resulting averages are depicted in the legend for each inquirer, and have been included in the table below for better reading. The table also includes the finished model F^+, which equals D^++320ms.

Inquirers	D^+Average	F^+Average
1	965ms	1285ms
2	48ms	368ms
3	11ms	331ms
4	10ms	330ms
5	10ms	330ms
6	10ms	330ms
7	10ms	330ms

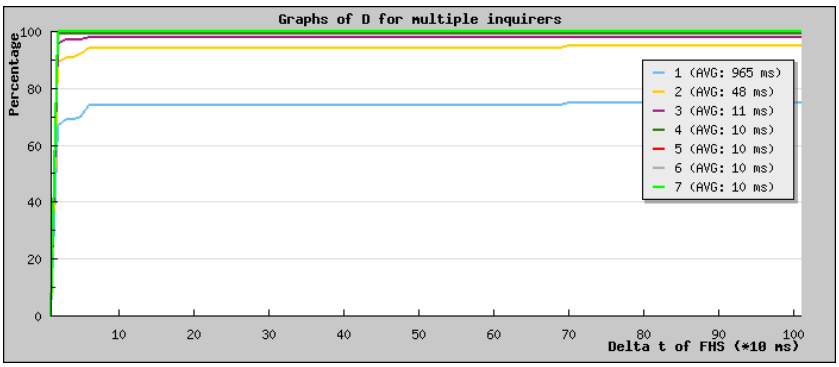


Figure 4.15: Cumulative pdf of $D^+$for multiple inquirers

D^-

When observing the graph of D^-, it is clear that this graph contains many shiftings of D from the positive to the negative domain. Due to its rigid nature it is not possible to easily extract information from the graph. However, this graph clearly models the behavior of D better than the graph with the non-negative D^+behavior. In this case, an average of around 80 percent of the devices is initially discovered, thus during the backoff time of the inquiry scanner. This conforms to the graph of F which also indicates that this is indeed the case. When following the average trend line of the graph, the following can be seen: At first, around 80 percent of the devices is discovered. The graph then shows no increase until around 150 milliseconds. There it increases until 250 milliseconds. This can best be seen by observing the outer extremities, or peaks, of the graph. The function than stabilizes until around 650 milliseconds where an incline is started which lasts until around 800 milliseconds.

The behavior of this graph, although rigid, does resemble the behavior of F, in the sense that the inclines are aligned at roughly the same position as in F, minus the 640 millisecond maximal backoff time. Figure 4.14 shows this more clearly as the average graph is included.

When using this rigid graph D^-, the resulting probabilities for multiple inquirers can be calculated, leading to the result of figure 4.16. Note that the scale of the x-axis is modified to exclude the large flat continuation of the graph for $\Delta t > 1000ms$. The resulting averages are depicted in the legend for each inquirer, and have been included in the table below for better reading. The table also includes the finished model F^-, which equals D^-+$320ms$.

Inquirers	D^-Average	F^-Average
1	560	880
2	-12	308
3	42	364
4	-1	319
5	10	330
6	10	330
7	10	330

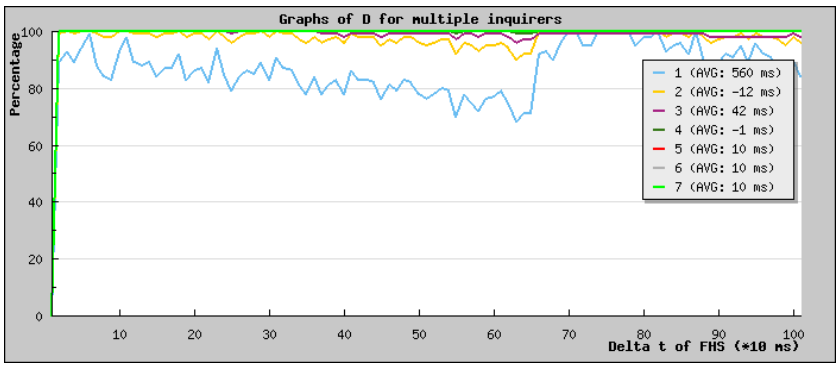


Figure 4.16: Cumulative pdf of $D^-$for multiple inquirers

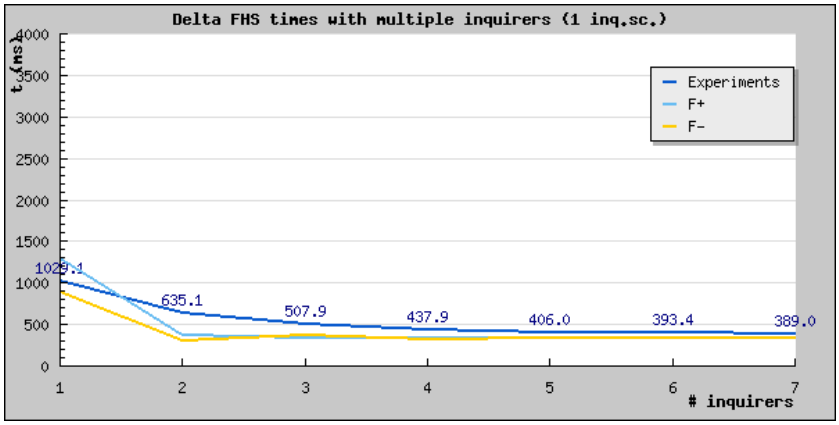


Figure 4.17: Average FHS interval times including $F^+$and F^-

4.3.4 Discussion

The model derived in this section and depicted in figure 4.17, is a crude model at best. Using the model, it is not possible to accurately predict the FHS interval times for multiple inquirers. In conclusion, no model can be derived from the original measurements by the ways explored in this section.

There are three reasons why this model might have failed:

- **Non-uniform behavior** If B in the used devices does not have the uniform distribution over 640ms that it should have according to the Bluetooth specification, the convolution will fail.

- ϵ **too large** If there is too much noise in the measurements, the convolution could fail. Because the convolution is recursive, it relies on previously calculated information. If a certain ϵ is introduced in an early phase, this will eventually blow up and become a major influence in the model.

- **Dependence** If both B and D are dependent, applying a convolution does not result in a valid probability distribution.

It is our estimation that too much noise in the measurements is not the problem. The graphs of F (figure 4.11), the average FHS interval times (figure 4.10 and the graph of the FHS delays (figure 4.3) show behavior that is to be expected. For noise to have the impact that is currently observed, it should have been pertinently present. Both the dependency and non-uniformity are most likely to be the source of the problem. The implementation of the Bluetooth standard on the chipset of the used devices (section 3.2) can not be verified. If the assumed uniform behavior of B is not the case, the results of the convolution could be very different. The Bluetooth specification states that when the used scan interval is smaller or equal to 1.28 seconds, the backoff time B can be either uniformly distributed among 0..1023 slots or 0..127 slots. Although 1023 slots is set to be the default, the manufacturer could have used the 127 option to possibly increase discovery times for his product. Nevertheless, this probably is not the case when the following observation is taken into account. Figure 4.3 shows the FHS delay of the new experiment. Here it can be seen that the assumption that all data has to be arrived at the $B + 11.25ms$ mark shows that this B is around 640 milliseconds. It also seems to be uniformly distributed, but unfortunately this can not be proven. If the manufacturer uses a badly seeded random number generator, or on purpose uses a non-uniform distribution (for example, only an even amount of slots), there still could be a problem. Although this might be the case, figure 4.3 shows that the distribution is probably at least semi-uniform. Finally there is the dependency assumption. If it is assumed that B and D are independent, the convolution should result in a valid model for the discovery time within the FHS intervals. The independency can be made comprehensible by realizing that the distribution of D does not depend on how long the backoff lasts. The way in which D depends on the Bluetooth specification regarding scan intervals and scan windows, supports this. If the convolution however fails like in figure 4.13, the influence of a possible dependency shows. A dependency, although unsuspected at first, can still exist. Because the convolution is a recursive process, the small error that will be introduced in the early phase can blow up to become a major influence in the model. The duration of B does influence the position of the timers of the Bluetooth chip. It could also influence other decision-making procedures that manufacturer could have implemented. Therefore, we can never be sure that there is no dependency in this particular used device. Future research should look into this independence assumption.

In the end this model, in the current state, can not be used to predict FHS intervals for multiple inquirers.

The research question states that the accuracy of the model is important. As discussed before the accuracy of this model is not good. A discussion between both models can be found in section 4.5. The scalability of this model however is very good, due to the fact that it is based on the same probabilistic modeling technique.

4.4 Related Work

[33] studies the multi-collision probability. This multi-collision is defined as an s-collision. It explains how the probability of collisions can be calculated using the birthday problem. As we also face a collision probability calculation

problem, and as discussed in section 4.2.3 the problem we face appears to the very same. Unfortunately this turned out not to be the case, because we have more information available about both colliding nodes. The birthday problem assumes that all nodes have the same behavior, which is not true for the Bluetooth discovery. One node keeps the same frequency for two periods of $1.28s$. The other node is scanning for this frequency, trying a frequency-train until a node is discovered. Because we know that not all devices behave the same way, the birthday paradox does not apply in its original form. We have used the reasons behind the paradox, resulting in a probability tree which fitted our problem.

In [5] a model is created for the statistical gate delay of Single and Multiple Input Switching models. The interesting part of this paper is its discussion on the convolution of probability density functions. In the paper the output arrival time is obtained by convoluting the gate delay and the input arrival time probability density functions. However, they conclude that due to suspected dependence of the gate delay this can not be done. This confirms that dependence is a valid reason when the convolution outcome does not match the measured outcome. We are not convinced that there is dependence in the functinos we convolve, but the results suggest a dependence possibility as mentioned in section 4.3.4.

In section 3.5 several papers are discussed which attempt a model of the discovery protocol. [26] and [10] model the discovery protocol using different approaches. The first uses a Matlab simulation to determine the performance of the process. The latter uses a custom built simulator. [16] uses a formal model of this process in the probabilistic modeling tool PRISM. Although some of the papers use a probabilistic model, they only do this to calculate a theorical model of the exact Bluetooth discovery protocol. In [7] exactly the same is done but this time using the VINT project network simulator. An approach containing probabilistic extrapolation of results from one to multiple devices is something we have not found in literature. To the best of our knowledge this approach is novice. We refer to section 3.5 for more information on the mentioned research.

Paper [34] studies the handover time at the MAC layer of wireless mobile networks, in particular of 802.11 (WLAN). A handoff occurs when a mobile station moves beyond the radio range of one access point, and enters in another coverage area. The so called "full scan handoff" scans for new stations using a discovery protocol. A model is suggested for the collision detection and avoidance, from which the first part is related to this research. Probability density functions are used to express the probability of a collision. Suppose a random variable X represents a collision per frame transmission, and X lies in range R. And R is limited by V_{MIN} and V_{MAX}. Then

$$X \in \{V_{MIN}, V_{MAX}\} where R \longrightarrow \{V_{MIN}, V_{MAX}\} \tag{4.23}$$

For calculating the number of collisions we use the same method, where X is the random variable denoting a collision, and R is the number of discovery channels. This collision scheme is then applied to every node in our graph 4.2.3 to calculate the number of transmissions, while R is corrected to the number of channels that are still available. They use a fixed number of channels while in our case it is adapted to the number of available frequencies.

[3] also applies probabilistic modelling to model parts of the MAC layer of 802.11. In particular the backoff time and timing related issues. Using the existing protocol, ways are found to decrease the implementation complexity for the model by identifying properties that can be used to simplify the model. [29] improves this by evaluating more optimizations to the model. Unfortunately, although the model is probabilistic, it is again modelled in PRISM for formal verification instead of prediction. Although the relation is there, the way of handling the actual model is different. Considering it is WLAN and not Bluetooth, the theoretical protocol-wise relation is minor. The same goes for [15] in which the CSMA/CD is protocol is modeled probabilistically, again in PRISM.

4.5 Conclusion

The research question:

> How accurate can the inquiry process be modeled using an empirical approach

can now be answered. The model using observation windows provides good accuracy for predicting the behavior of the system. As this is done using an empirical approach, the result is that it is indeed possible to model the inquiry process accurately using an empirical approach. The second model using FHS interval times however did not perform well.

The subquestion of scalability is equal for both approaches. Both approaches provide a scalable solution for extension across multiple inquirers.

5 Conclusions

This chapter contains the conclusions of the research, ordered by the research questions listed in section 1.4.

Note that the conclusions based on measurements are based on one type of device only (see section 3.1). Although the protocol, version 1.2, is the same for each device, small differences might still occur. To verify whether these conclusion are valid for other devices is left as future work.

5.1 Bluetooth behavior

- **How do multiple inquirers influence the discovery time for each inquiry scanner**
  with an arbitrary number of inquiring devices, between a certain time t and $t + \Delta t$, the same percentage of discoverable inquiry scanners within range is discovered. Provided the distance of the inquirers and the inquiry scanners is the same and the same conditions apply. The number of inquiry scanners within range is therefore of no importance in this relation. It has to be noted that this conclusion is limited to up to 20 inquiry scanners, the maximum of the experiment. This result can be used for predicting the number of inquiry scanners based on the number of detected devices in a given time.

The subquestions:

- **How many devices can be discovered**
  One inquirer can, without unrelated delay, detect at least 20 devices. Because of hardware limitations and because this value is acceptable for the purpose of this research, no test was performed using more than 20 devices. Based on the results of section A.2, in which it is measured that on average 19% of the people carry a bluetooth device, this currently enables handling crowds of over 100 people on average.

- **How much time is required to find all devices in an area**
  One inquiry scanner will, in a 6 second custom dutycycle find only 90% of the devices. Having more than one inquirer increases this percentage

to 100%. The average time to discover all devices is 5 seconds. If the distance between inquirers and inquiry scanners is 12 meters of more, the 100% will not be reached. In this case it reaches 95% in 6 seconds.

- **What is the ideal dutycycle for continuous scanning**
  The best dutycycle for continuous scanning is (6,7,8). Most devices will be found, and backoff time is limited.

- **What is the optimal number of inquirers**
  The optimal number of inquirers for minimal discovery time is 7. Having more will probably have a positive effect on the discovery time, but as can be seen in figure 4.1 the gain will decrease.

- **Is there a competition effect among inquirers**
  Yes. Although the maximum amount of 7 inquirers does not provide enough proof for establishing a measure for the competition effect in this research, an infinite number of inquirers would lead to zero discovery. Figure 4.7 shows the impact of this effect.

- **Is there information in measurements related to distance**
  From section A.1 it can be concluded that the RSSI value can be used to estimate distance. A model in the form of an equation has been established for the room in which the experiments have been performed.

5.2 Modeling the inquiry process

- **Can the inquiry process be accurately modeled**
  Overall, modeling based on a practical instead of theoretical basis is possible. Two models have been created. The model using observation windows approaches the real situation quite well, with errors ranging from 6% to 42%. This is the upmost accuracy we have obtained. The model using FHS interval times is not accurate and can not be used to make any predictions.

  Although literature studies acknowledge the difficulty of creating models analytically, this research shows that it is possible. In retrospect, it is not even that difficult when using a practical approach.

  The scalability of the models is good. With limited resources the model can be calculated for many inquirers.

5.3 Future work

There are several things which did not fit in the scope of this research, but require further investigation.

- **Handling over 20 devices** The possibility of handling over 20 inquiry scanners at once should be investigated. This would increase the size of the crowd that can be handled by the localization system. It would also be possible to establish whether the conclusions of this research are valid for these larger amounts.

- **Distance configuration** In this research the inquirers and inquiry scanners have been separated into two groups and set several distances apart. It would be very interesting to see what happens when both groups are interleaved at multiple distances. By placing different inquirers and different inquiry scanners at different semi-random distances, more information can be gathered of the behavior.
- **Behavior of other devices** The experiments of this research have been done using one particular Bluetooth device. The implementation and design choices may differ for each manufacturer. This means for example backoff times, scan intervals, differences in antennae, etcetera, can be implemented differently. Whether such devices display different behavior, and whether this influences the results of this research should be looked in to.
- **Observation window** In the discussion of the first model (section 4.2.4) an issue arises with the observation windows. The model does only partially take the backoff-time of the inquirers into account. The model is based on measurements for 1 inquirer. This does incorporate the backoff time for 1 inquirer. Whether an extra correction for backoff times of multiple inquirers should be performed is subject to further research.
- **B-D Independence assumption** The discussion of the second model presents an unforeseen problem. B and D do not seem to be independent. Whether this is indeed the case, and how they are related can be very interesting. It is therefore recommended to look into that.

Bibliography

[1] Bluez project. http://www.bluez.org/, 2009.

[2] Php: Hypertext preprocessor. http://www.php.net/, 2009.

[3] Amitabha Roy 0002 and K. Gopinath. Scalable probabilistic models for 802.11 protocol verification. *CoRR*, cs.LO/0403044, 2004.

[4] ASUS WL 500g Premium. http://www.asus.com/product.aspx?P_ID=8el2DcrRjLoHNdQ8&templete=2, 2009.

[5] Aseem Agarwal, Florentin Dartu, and David Blaauw. Statistical gate delay model considering multiple input switching. In *DAC '04: Proceedings of the 41st annual Design Automation Conference*, pages 658–663, New York, NY, USA, 2004. ACM.

[6] Mortaza S. Bargh and Robert de Groote. Indoor localization based on response rate of bluetooth inquiries. In *MELT '08: Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, pages 49–54, New York, NY, USA, 2008. ACM.

[7] Stefano Basagni, Raffaele Bruno, and Chiara Petrioli. Device discovery in bluetooth networks: A scatternet perspective. In *NETWORKING '02: Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications*, pages 1087–1092, London, UK, 2002. Springer-Verlag.

[8] H. Brakman, V. Driessen, J. Kavuma, L. Nij Bijvank, and S. Vermolen. Formal model of the bluetooth inquiry protocol. 2006.

[9] Debasish Chakraborty, Goutam Chakraborty, Sagar Naik, and Norio Shiratori. Discovery and delay analysis of bluetooth devices. In *MDM '06: Proceedings of the 7th International Conference on Mobile Data Management*, page 114, Washington, DC, USA, 2006. IEEE Computer Society.

[10] Goutam Chakraborty, Kshirasagar Naik, Debasish Chakraborty, Norio Shiratori, and David Wei. Analysis of the bluetooth device discovery protocol. *Wireless Networks*, 2008.

[11] Yu chung Cheng, Yatin Chawathe, Anthony Lamarca, and John Krumm. Accuracy characterization for metropolitan-scale wi-fi localization. In *In Proceedings of Mobisys 2005*, pages 233–245, 2005.

[12] Aditus Consulting. Jpgraph. `http://www.aditus.nu/jpgraph/`, 2009.

[13] Don Coppersmith. Another birthday attack. In *Lecture notes in computer sciences; 218 on Advances in cryptology—CRYPTO 85*, pages 14–17, New York, NY, USA, 1986. Springer-Verlag New York, Inc.

[14] DD-WRT. `http://www.dd-wrt.com/`, 2009.

[15] M. Duflot, L. Fribourg, T. Hérault, R. Lassaigne, F. Magniette, S. Messika, S. Peyronnet, and C. Picaronny. Probabilistic model checking of the CSMA/CD protocol using PRISM and APMC. In *Proc. 4th Workshop on Automated Verification of Critical Systems (AVoCS'04)*, volume 128(6) of *Electronic Notes in Theoretical Computer Science*, pages 195–214. Elsevier Science, 2004.

[16] Marie Duflot, Marta Kwiatkowska, Gethin Norman, and David Parker. A formal analysis of bluetooth device discovery. In *In Proc. 1st International Symposium on Leveraging Applications of Formal Methods (ISOLA04)*, 2004.

[17] E. Elnahrawy, Xiaoyan Li, and R.P. Martin. The limits of localization using signal strength: a comparative study. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 406–414, Oct. 2004.

[18] Sahar Idwan, Suad Alramouni, Mosleh Al-Adhaileh, and Ahmad Al-Khasawneh. Enhancing mobile advertising via bluetooth technology. *Int. J. Mob. Commun.*, 6(5):587–597, 2008.

[19] Apple Inc. iphone and ipod touch: Supported bluetooth profiles. `http://support.apple.com/kb/HT3647`, 2009.

[20] Hewlett-Packard Invent. Wi-fi and bluetooth - interference issues, 2002.

[21] Prof. David Joyner. Mysql. `http://www.mysql.com`, 2009.

[22] Jeffrey P. Kharoufeh. Bluetooth inquiry time characterization and selection. *IEEE Transactions on Mobile Computing*, 5(9):1173–1187, 2006. Member-Peterson, Brian S. and Senior Member-Baldwin, Rusty O.

[23] Hyuk Lim, Lu-Chuan Kung, Jennifer Hou, and Haiyun Luo. Zero-configuration indoor localization over ieee 802.11 wireless infrastructure. *Wireless Networks*, 2008.

[24] C. J. Escudero O. Fresnedo, D. Iglesia. Bluetooth inquiry procedure: Optimization and influence of the number of devices. In *International Conference Communication Systems and Networks (IASTED-CSN)*, pages 29–31, Palma de Mallorca, Spain, 2007.

[25] Veljo Otsason, Alex Varshavsky, Anthony Lamarca, and Eyal de Lara. *Accurate GSM Indoor Localization.* 2005.

[26] Brian S. Peterson, Rusty O. Baldwin, and Richard A. Raines. Bluetooth discovery time with multiple inquirers. In *HICSS '06: Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, page 232.1, Washington, DC, USA, 2006. IEEE Computer Society.

[27] Network Time Protocol. `http://www.ntp.org/`, 2009.

[28] Matthias Ringwald and Kay Romer. Practical time synchronization for bluetooth scatternets. In *Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on*, pages 337–345, Sept. 2007.

[29] Amitabha Roy and K. Gopinath. K.: Improved probabilistic models for 802.11 protocol verification. In *In: Proc. 17th International Conference on Computer Aided Verification (CAV05), LNCS*, pages 239–252. Springer, 2005.

[30] O. Sasaki and T. Akiyama. Multipath delay characteristics on line-of-sight microwave radio system. *Communications, IEEE Transactions on*, 27(12):1876–1886, Dec 1979.

[31] Bluetooth Special Interest Group (SIG). *Specification of the Bluetooth System (Core 2.1+EDR).* 2007.

[32] Timothy J. Smith, Stefan Saroiu, and Alec Wolman. Bluemonarch: a system for evaluating bluetooth applications in the wild. In *Mobisys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 41–54, New York, NY, USA, 2009. ACM.

[33] Kazuhiro Suzuki, Dongvu Tonien, Kaoru Kurosawa, and Koji Toyota. Birthday paradox for multi-collisions. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E91-A(1):39–45, 2008.

[34] Aasia Riasat Syed S. Rizvi and Khaled M. Elleithy. A quantitative analysis of handover time at mac layer for wireless mobile networks. *International Journal of Wireless and Mobile Networks (IJWMN)*, 1(2), 2009.

[35] T. Thamrin and S Sahib. The inquiry and page procedure in bluetooth connection. *International conference of Soft Computing and Pattern Recognition*, 2009.

[36] Wikipedia. `http://en.wikipedia.org/wiki/Global_Positioning_System#Selective_availability`, 2009.

[37] Wikipedia. `http://en.wikipedia.org/wiki/USB_hub#Protocol`, 2009.

[38] Wikipedia. `http://en.wikipedia.org/wiki/Universal_Serial_Bus`, 2009.

[39] M. Wright, D. Stallings, and D. Dunn. The effectiveness of global positioning system electronic navigation. In *SoutheastCon, 2003. Proceedings. IEEE*, pages 62 – 67, 4-6 2003.

A

Miscellaneous

This chapter contains all miscellaneous information on this thesis. The first section discusses the effect of distance that can be seen after the experiments have been completed. The second part discusses an experiment conducted to count the number of people that actually have their Bluetooth enabled. The third section is about the tools which were designed in order to take the measurements and present them to the user.

A.1 Effect of distance in the experiment

This section discusses the two effects of distance that have been measured in the experiments. They are, in order, discovery time and signal strength.

Discovery time

Figure 3.14 shows the general results of the experiment with different distances. Two graphs have been selected which represent the general behavior. It can be seen that generally speaking the following condition holds; when the devices are further away, their detection takes longer. Although this may not seem surprisingly, it can be seen in the graphs that there are relations between detection and distance which occur more frequently. Looking closely at both graphs, it can be seen that the amount of space between the individual lines of each graph is of a corresponding nature. Therefore this could potentially be used to determine the relative distance of an object as it moves away.

The reason for this delayed detection at increased distance can be explained fairly easy. A wrong assumption might be that because the devices are further apart, the signal requires more line-of-sight time to reach the inquirer. As radio signals travel at the speed of light, this will have no influence on a millisecond scale. The reason why the inquiry takes more time is twofold:

- **Noise** As distance increases, the signal strength decreases (section A.1). This makes the signal more vulnerable to interference.
- **Reflection** As distance increases, so does the influence of reflection problems such as multipath fading. To correct for such effects, the Bluetooth

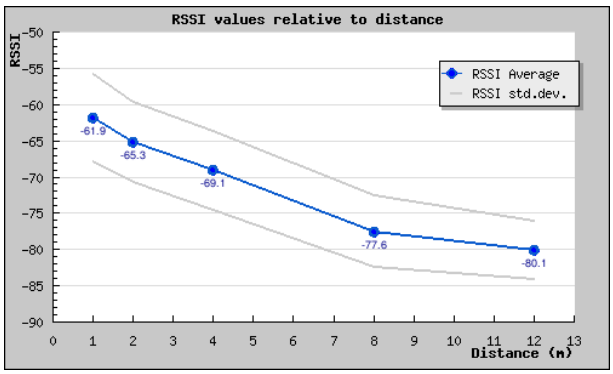
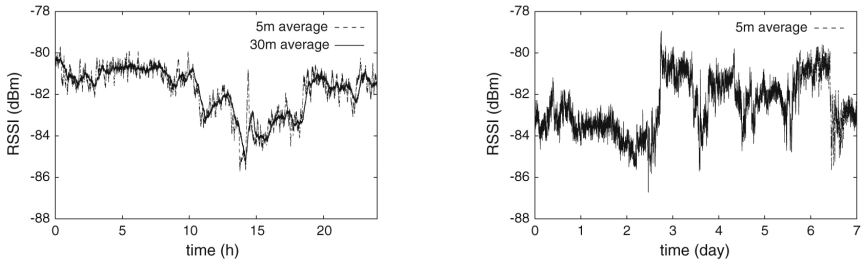


Figure A.1: RSSI values relative to distance



(a) Signal strength variations for one day

(b) Signal strength variations for one week

Figure A.2: RSSI deviation over time [23]

devices are often equipped with hardware to counter this problem. As the reflections introduce noise and take arbitrary time to reach the devices, a small delay may take place [30].

Signal strength

Figure 3.14 shows the general results of the experiment with regard to different distances. It can be seen that the RSSI values decrease as the distance increases. The Bluetooth specification [31] states that a semi-random deviation of +/- 6dB is possible possible and acceptable for the reported RSSI values by the hardware. The graphs also shows the RSSI standard deviation, drawn in gray above and below the blue average-line. The standard deviation of this blue line is around 6dB, which can therefore be explained partially by the Bluetooth standard. The average standard deviation of all performed experiments is 4.9588. Also small environmental changes and changes in condition may impact the RSSI values of the single measurements. Research has shown that the signal variations that occur "naturally", can be considerable [23]. Figure A.2 shows the variations in RSSI which have occurred in [23] without any deliberate physical changes.

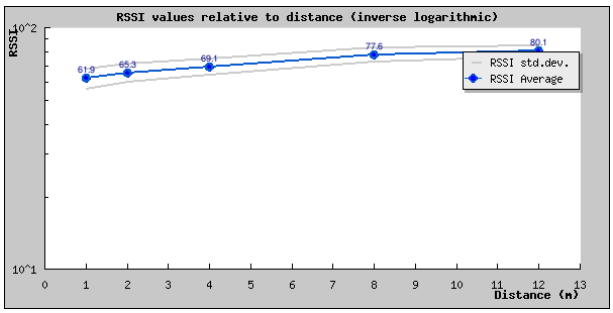


Figure A.3: RSSI values relative to distance, positive logarithmic scale

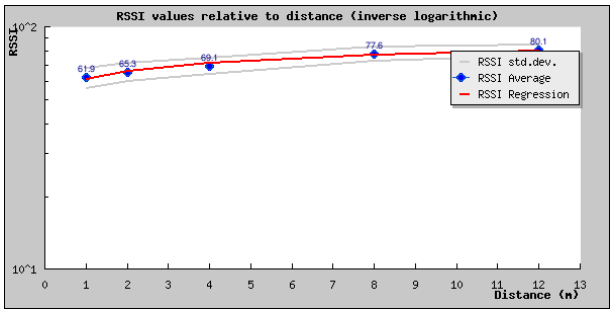


Figure A.4: RSSI regression analysis (equation A.4), positive logarithmic scale

Figure A.3 shows graph A.1 on a positive logarithmic scale. As the graph shows almost linear logarithmic behavior, this relation could be used for estimating distance in the given environment. The linear function can be derived using regression analysis; an adapted version of the method of least square fitting to find the coefficients:

$$y = a + b \ln x \tag{A.1}$$

$$b = \frac{n \sum_{i=1}^{n} (y_i \ln x_i) - \sum_{i=1}^{n} y_i \sum_{i=1}^{n} (\ln x_i)}{n \sum_{i=1}^{n} (\ln x_i)^2 - (\sum_{i=1}^{n} \ln x_i)^2} \tag{A.2}$$

$$a = \frac{n \sum_{i=1}^{n} y_i - b \sum_{i=1}^{n} (\ln x_i)}{n} \tag{A.3}$$

This results in the following equation for the RSSI value with respect to distance:

$$y = 60.65 + 7.64 \ln x \tag{A.4}$$

Figure A.4 shows the original data points of the average RSSI line in combination with the regression analysis.

A.2 Crowd scanning

The focus of the BlueWhere project lies on tracking people as they move around. The amount of people that have Bluetooth enabled on one of their

devices is of great importance. If there are too few people that have a discoverable device, the system would not satisfy its customer. On the other hand, if too many people have their Bluetooth turned on, it might lead to a different problem. In that case it might be possible that the system gets overloaded and accuracy may no longer be guaranteed.

In order to find out how many people have a Bluetooth discoverable device, a field test has been performed. On an average week-day, between 16:00 and 17:00 hours, several locations in the city of Enschede (NL) were subject to this test. A simple tool has been created and used to inquire and store the data to a portable flash drive.

Location	People	Bluetooth	Percentage
A	50	7	14%
B	20	6	30%
B	20	5	25%
C	20	1	5%
C	20	1	5%
D	40	12	30%
D	40	11	27.5%
E	20	1	5%

Table A.1: Bluetooth enabled people

The locations on which the tests are performed, need to be carefully selected. It must be possible to estimate the amount of people that are in range. Also, not too many people must enter or leave this range while inquiring. To ensure that different types of people are included in this test, the locations differ. The entrance of a supermarket, electronics store and a pub are a means of making sure that different consumer groups were included.

Table A.1 shows the values of the eight tests. The letters represent different locations on which the test has been performed. The second column shows an estimation of the amount of people the were in Bluetooth range, followed by the amount of discovered devices and the derived percentage.

On average, among 230 people 44 devices have been discovered, or 19%. Although the deviation among the experiments is large (11.7), it can be concluded that people indeed carry discoverable Bluetooth devices. Furthermore, these devices are set to "discoverable" making them useful for the BlueWhere project.

The results of this simple test are not reliable. The conclusion is based on the assumption that a person only carries up to one device. In the end, the test does however provide an insight in the availability of these devices on the street.

A.3 Tools

This section gives an overview of the tools that have been developed for this research.

Automated measuring tool

The automated measuring tool was created to assist in doing the initial experiments. An experiment consists of a few basic parameters:

- Dutycycle
- How many (possibly which) inquirers to use
- How many (possibly which) inquiry scanners to use
- Power level of the devices
- Distance between inquirers and inquiry scanners

After defining an experiment, it needs to actually run and the received FHS packets should be stored in a database. Doing this by hand would create an unfeasible amount of work. The created tool is written in C and has a commandline interface. It thus receives the parameters of the experiment via the commandline, and runs the experiment accordingly.

Usage

```
Usage: ./BluetoothTool [OPTIONS]...
Perform Bluetooth inquiry with given parameters.

Arguments:
  -w    Inquiry window (default: 4)
  -m    Inquiry period minimal (default: 6)
  -M    Inquiry period maximal (default: 7)
  -i    List of inquirers (default: none)
  -s    List of inquiry scanners (default: none)
  -p    Powerlevel of inquirers from -8 to 0 (default: 0)
  -n    Number of inquiry experiments (default: 10)
  -X    Number of the experiment
  -f    Force removal of experiment if it already exists.
  -d    Distance of the scanners (For DB storage only!)
  -c    Comment for the experiment (For DB storage only!)
  -o    Filename of the output storage. SQL is disabled if
      this option is used!
  -h    Display this help and exit

Exit status is 0 if OK, other value in case of problems.

Report bugs to <anne.franssens|robert.degroote>@novay.nl
```

The first three parameters define the dutycycle. A regular (6,7,8) would thus be entered as `-w6 -m7 -M8`. The list of inquirers and inquiry scanners can be specified in two different ways. A list of single devices can be specified as a comma-separated list (`-i label1,label2,label3`), or a range of devices can be entered (`-i label1,label{5-12}`). The number of dutycycles each inquirer has to complete can be specified using the `-n` argument. The next four arguments are used as metadata for the experiment, and stored in the database accordingly. The `-o` argument allows for the specification of an output file. If

this file is specified, the FHS information will be placed in comma-separated-value files instead of in the database.

When several experiments need to be performed in sequence, a shell-script or batch-file can be created. Such a script would typically contain calls to the tool:

```
./BluetoothTool -w6 -m7 -M8 -d8 -X1077 -n250 -i BW{49-55} -s BW{65-74}
./BluetoothTool -w6 -m7 -M8 -d8 -X1078 -n250 -i BW{49-55} -s BW{65-77},BW56,BW58
```

Design

The application is split into four parts:

- Main (main.c)
- Bluetooth control (btcontrol.c)
- MySQL extension (mysql_ext.c)
- File extension (file_ext.c)

the main file of the application takes care of handling the arguments and the general execution of the experiment. It does this by using functions from the other files. The Bluetooth control contains all functions that interface the BlueZ stack. The MySQL extension contains an interface to the MySQL client library. The file extension contains an interface to writing the comma-separated-value files. The functions that show the operation of the files can be found in appendix C.

Web service

In order to provide an insight into the gathered data, a web service has been developed using PHP [2]. This service reads from the database tables and presents it to the user accordingly. It makes it easy to see which experiments are in the database, what the parameters were, and their basic results and analysis. The web service is split into three parts; the index with the overview of all experiments, a web page for every experiment with basic information, and some graphs created with the framework.

Figure A.5 shows an example of the index with the overview of all experiments. The basic parameters of the experiments can be seen, which makes it easy to navigate through the extensive list. Every row of the table can be selected in order to proceed to the detailed information on the experiment. The detailed view of every experiment contains a few basic elements:

- **List of discovered devices** Figure A.6. This contains, for every discovered device, the details about this device including the manufacturer. Information about the device in the experiment is also available, such as average RSSI value and deviation, total number of received FHS packets and discovery times.
- **FHS packets per dutycycle** Figure A.7. This graph shows the number of FHS packets that are received per dutycycle.

Bluetooth™ Experiments

Index

- Analysis of dutycycle experiments
- inquirer dependency graph
- RSSI Analysis

Table 1: Experiments

EXPERIMENTS	DUTY CYCLE	POWER LEVEL	INQUIRIES	INQUIRERS	INQUIRY SCANNERS	DISTANCE	START	END	COMMENT
8	20-21-22	0	100	1	5	2	2008-12-03 13:04:07	2008-12-03 13:50:07	Experiment 4, 1 inquirer, 5 scanners
9	20-21-22	0	100	1	10	2	2008-12-03 13:50:48	2008-12-03 14:36:45	Experiment 4, 1 inquirer, 10 scanners
10	20-21-22	0	100	1	15	2	2008-12-03 14:37:28	2008-12-03 15:23:23	Experiment 4, 1 inquirer, 15 scanners
11	20-21-22	0	100	1	20	2	2008-12-03 15:24:06	2008-12-03 16:10:09	Experiment 4, 1 inquirer, 20 scanners
13	20-21-22	0	100	3	5	2	2008-12-03 16:10:53	2008-12-03 16:56:49	Experiment 4, 3 inquirers, 5 scanners
14	20-21-22	0	100	3	10	2	2008-12-03 16:57:31	2008-12-03 17:43:37	Experiment 4, 3 inquirers, 10 scanners

Figure A.5: Web service, screenshot of index

DEVICE	LABEL	MAC ADDRESS	MANUFACTURER	FHS PACKETS	RSSI AVERAGE	RSSI VARIANCE	RSSI STD. DEVIATION	FIRST DISCOVERY	LAST DISCOVERY
29	BW69	00:09:DD:50:23:64	Mavin Technology Inc.	4959	-67.5	25.2	5	2008-12-20 03:51:29	2008-12-20 04:39:23
31	BW68	00:09:DD:50:23:83	Mavin Technology Inc.	4834	-69.3	21.2	4.6	2008-12-20 03:51:28	2008-12-20 04:39:23
33	BW65	00:09:DD:50:23:56	Mavin Technology Inc.	5083	-66.2	24.9	5	2008-12-20 03:51:28	2008-12-20 04:39:22
35	BW66	00:09:DD:50:20:AF	Mavin Technology Inc.	4779	-65.4	22.5	4.7	2008-12-20 03:51:28	2008-12-20 04:39:22
40	BW67	00:09:DD:50:22:00	Mavin Technology Inc.	4886	-67.5	18.4	4.3	2008-12-20 03:51:28	2008-12-20 04:39:20

Figure A.6: Web service, screenshot of experiment, discovered devices

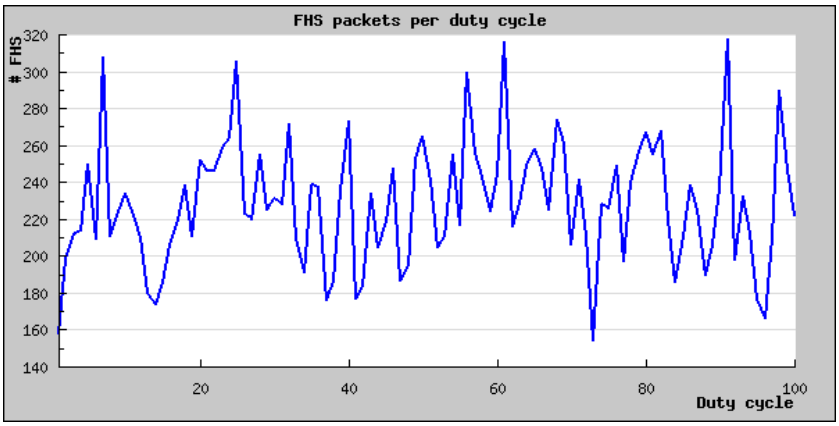


Figure A.7: Web service, screenshot of experiment, fhs packets

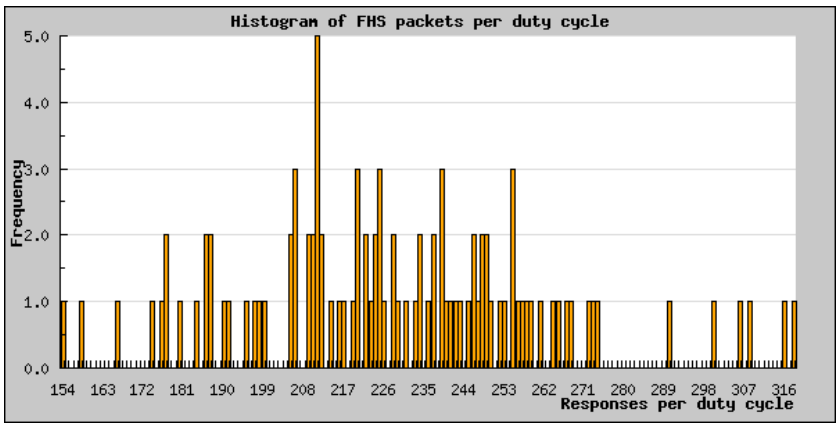


Figure A.8: Web service, screenshot of experiment, fhs histogram

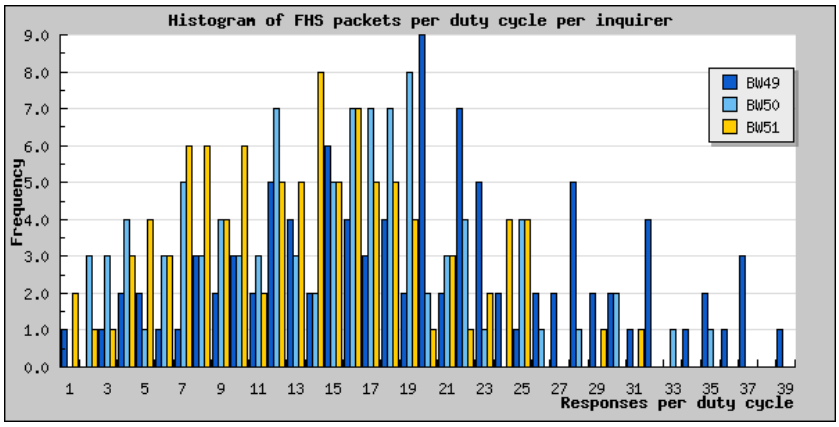


Figure A.9: Web service, screenshot of experiment, combined fhs histogram

- **Histogram of FHS packets per dutycycle** Figure A.8. This graph shows a histogram of the received FHS packets per dutycycle. The horizontal axis shows the amount of FHS responses, the vertical axis how often this amount of FHS responses has occurred in a dutycycle.
- **Histogram of FHS packets per dutycycle per inquirer, overview** Figure A.9. This graph shows the same information as the previous graph, except for that it is on a per inquirer basis.
- **Histogram of FHS packets per dutycycle per inquirer** figure A.10. For each inquirer there is a histogram containing the same data as in the previous graph.
- **Correlation of histograms per inquirer** Figure A.11. The previous graphs show a histogram for each inquirer. How these graphs correlate is shown in this figure. This way it can be determined if there is correlation of the received amount of FHS packets of different inquirers.

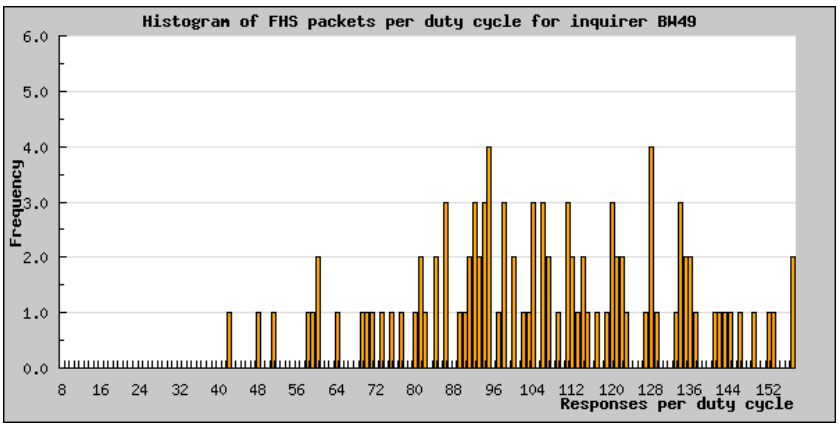


Figure A.10: Web service, screenshot of experiment, fhs histogram per inquirer

DEVICES	BW51	BW50	BW49
BW51		0.63	0.83
BW50			0.73
BW49			

Figure A.11: Web service, screenshot of experiment, inquirer correlation

Data analysis

The analysis of the data is mostly done by SQL queries. Although it could have been done using regular programming, filtering the data using well designed queries increases performance of the analysis. An example of this is the *custom dutycycle*. The custom dutycycle is a concept which subdivides all data into segments of predetermined duration. Whereas usually the dutycycle is used, now the custom dutycycle has to be used. As all analysis is done on a per-dutycycle basis, having SQL calculate the custom dutycycle is an important advantage. When using this division in a subquery, the resulting view of the database can then be queried as if the custom dutycycle were the actual dutycycle.

```
1        SELECT
2        id,
3        device_id,
4        floor((UNIX_TIMESTAMP(datetime)*1000+milliseconds-
             $datetime_start)/$cycle) as `customdutycycle`,
5        MIN(UNIX_TIMESTAMP(datetime)*1000+milliseconds)-
             $datetime_start-($cycle*floor((MIN(UNIX_TIMESTAMP
             (datetime)*1000+milliseconds)-$datetime_start)/
             $cycle)) as datetime
6        FROM measurements WHERE experiment_id=$experiment_id
             where GROUP BY `customdutycycle`, device_id
```

The graphs are created using JpGraph [12], a free library that provides support for creating graphs using PHP. The graphs for this thesis were also created using that library.

B

MySQL tables

Table B.1: Structure of table crowdscanner

Field	Type	Comment
date	datetime	Time of the measurement
peoplecount	int	Amount of people in the vicinity
mac	string	MAC address of the found device
class1	int	Bluetooth major service class
class2	int	Bluetooth major device class
class3	int	Bluetooth minor device class

Table B.2: Structure of table devices

Field	Type	Comment
id	int	ID of the entry
label	string	Name of the device, as on its label
address	string	MAC address of device
major_service_class	int	Bluetooth device class
major_device_class	int	Bluetooth device class
minor_device_class	int	Bluetooth device class
comment	string	Comments

Table B.3: Structure of table dutycycles

Field	Type	Comment
id	int	ID of the entry
experiment_id	int	ID of the experiment (table experiments)
inquirer_id	int	ID of the inquirer device (table devices)
dutycycle	int	Number of the dutycycle
datetime	datetime	Date and time of the entry

Table B.3: Structure of table dutycycles (continued)

Field	Type	Comment
milliseconds	int	Milliseconds of datetime

Table B.4: Structure of table experiments

Field	Type	Comment
id	int	ID of the experiment
inquiry_window	int	Dutycycle inquiry window time
inquiry_period_min	int	Dutycycle minimal inquiry period time
inquiry_period_max	int	Dutycycle maximal inquiry period time
powerlevel	int	Power setting of the devices
inquiries	int	Number of inquiries done
inquirers_cnt	int	Number of used inquirers
inquiry_scanners_cnt	int	Number of used in inquiry scanners
distance	int	Distance between inquirers and inquiry scanners
date_start	datetime	Start of the experiment
date_end	datetime	End of the experiment
command	string	Shell command which initiated the experiment
comment	string	Comments

Table B.5: Structure of table manufacturers

Field	Type	Comment
id	int	ID of the manufacturer
mac	string	First 6 hex characters that identify this manufacturer
company	string	Manufacturer name
address	string	Address of the manufacturer

Table B.6: Structure of table measurements

Field	Type	Comment
id	int	ID of the measurement
experiment_id	int	ID of the experiment (table experiments)
inquirer_id	int	ID of inquirer (table devices)
dutycycle	int	ID of dutycycle (table dutycycles)
device_id	int	ID of found inquiry scanner (table devices)
rssi	int	RSSI value of discovery
datetime	datetime	Date and time of the entry
milliseconds	int	Milliseconds of datetime

C

Functions of automated measuring tool

Sourcecode C.1: Functions of main.c

```
1  //Print help when called with -h argument
2  void printHelp(char *);
3
4  //Processes the list of inquirers from commandline
5  int handle_commandline_i(char *argument);
6  //Processes the list of inquiry scanners from commandline
7  int handle_commandline_s(char *argument);
8
9  //Assists handle_commandline_x(..)
10 int _parse_Commandline_Devices(char deviceList[][
       MAX_DEVICE_LABELSIZE], char *argument);
11
12 //Shuts down all devices that are not used
13 int bringDownUnusedDevices();
14
15 //Prints debug information
16 int print_hciconfig();
```

Sourcecode C.2: Functions of btcontrol.c

```
1  //Initialize library
2  void init();
3
4  //Singular device functions
5  int openDevice(int deviceId);
6  int resetDevice(int deviceId);
7  int closeDevice(int deviceId);
8
9  //Bring device(s) up/down
10 int initAllDevices();
11 int bringUpDevice(int deviceId);
12 int bringDownDevice(int deviceId);
13 int bringUpAllDevices();
14 int bringDownAllDevices();
15
```

```
16 //Check status
17 int isUp(int deviceId);
18 int isDown(int deviceId);
19
20 //List devices
21 int listAllDevices(int *deviceList);
22 int listAllUpDevices(int *deviceList);
23 int _listAllDevices(int *deviceList, int flag_up);
24
25 //Retrieve a device's bluetooth address
26 int readLocalAddress(int deviceId, char *pAddress);
27 //retrieve device ID by address
28 int getIdFromAddress(char *pAddress);
29
30 //Power levels
31 int setTransmitPowerLevel(int deviceId, int powerLevel);
32 int getTransmitPower(int deviceId);
33
34 //Enable/disable inquiry scan
35 int enableInquiryScan(int deviceId, bool bEnableInquiryScan);
36 int startPeriodicInquiry(int deviceId, int window, int
       minperiod, int maxperiod);
37 int stopPeriodicInquiry(int deviceId);
38
39 //Check if there are pending inquiry results for device with
       ID deviceId
40 int pollInquiryResults(int deviceId);
41 int getNumberOfInquiryResults(int deviceId);
42 void getInquiryResult(int deviceId, int index, inquiryResult
       *pResult);
43 void resetInquiryResult(int deviceId);
44
45 //Reset all devices
46 int resetAllDevices();
```

Sourcecode C.3: Functions of mysql_ext.c

```
1  //Connect to sql
2  int sql_connect();
3
4  //Store information into tables
5  int sql_addWindow(int experiment_id, int inquirer_id, int
       windowsize, int dutycycle);
6  int sql_addFHS(int experiment_id, int host_device_id, int
       inquirywindow, char *address, int rssi, time_t time, long
        int time_msec);
7  int sql_addExperiment(int argc, char **argv, int
       experiment_id, int inquiry_window, int inquiry_period_min
       , int inquiry_period_max, int powerlevel, int inquiries,
       int inquirers_cnt, int inquiry_scanners_cnt, int distance
       , char *comment);
8  int sql_endExperiment(int experiment_id);
9  int sql_new_experiment_id();
10
11 //Check if experiment is in db already
```

```
12 int sql_experiment_exists(int experiment_id);
13
14 //Device DNS functions
15 int sql_get_address_by_label(char *label, char *address);
16 int sql_get_device_id(char *address);
17 int sql_print_internalDeviceList();
18
19 //Remove experiment from tables
20 int sql_experiment_remove(int experiment_id);
21
22 //Disconnect sql
23 int sql_disconnect();
```

Sourcecode C.4: Functions of file_ext.c

```
1  //Open files
2  int file_open(char *filename);
3
4  //Store information into files
5  int file_addExperiment(int argc, char **argv, int
       experiment_id, int inquiry_window, int inquiry_period_min
       , int inquiry_period_max, int powerlevel, int inquiries,
       int inquirers_cnt, int inquiry_scanners_cnt, int distance
       , char *comment);
6  int file_addWindow(int experiment_id, char *inquirer_address,
        int windowsize, int dutycycle);
7  int file_addFHS(int experiment_id, char *host_device_address,
        int inquirywindow, char *address, int rssi, time_t time,
        long int time_msec);
8
9  //Ends the experiment
10 int file_endExperiment();
11
12 //Close the files
13 int file_close();
```