Fast and Energy Efficient Neighbour Discovery for Opportunistic Networking with Bluetooth

by

Cătălin Drulă

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

0-494-07171-0

Your file Votre référence
ISBN:
Our file Notre reterence
ISBN:

Abstract

Fast and Energy Efficient Neighbour Discovery for Opportunistic Networking with Bluetooth

Cătălin Drulă

Master of Science

Graduate Department of Computer Science

University of Toronto

2005

This dissertation proposes a novel scheme for neighbour discovery in Bluetooth that achieves fast discovery times and low power consumption. To tune the parameter values of the scheme, we develop a simulator to comprehensively explore the large parameter space and then validate the simulation results for selected parameter values through measurements with real devices. Our results show a two-fold improvement in both mean discovery time and power consumption, at the same time, over the previously best known scheme.

The evaluation of our neighbour discovery scheme reveals that there is a trade-off in the selection of its parameters between energy efficiency and discovery time. In the second part of this thesis, we propose two adaptive algorithms for dynamically switching these parameters, based on past activity. We evaluate these algorithms in a node mobility simulation. Our adaptive algorithms reduce energy consumption by 50% and have up to 8% better performance over a naïve, power-conserving scheme.

Acknowledgements

First foremost, I would like to thank my supervisor, Cristiana Amza. Without her constant encouragement and support this thesis would not have existed.

I cannot describe in words the support that my parents have been for me over these two years. Thanks for your kind words, thanks for being there for me, thanks to my father for encouraging me to undertake this Master's. I also want to thank my brother for the countless times he put up with my complaining and the huge phone bills I have caused him.

I also want to thank everybody I have worked with at the LSR-IMAG laboratory in France during my exchange year there: Franck, Andrzej, Yan, Martin. Thanks guys for a great year.

Finally, a big thank you to all my friends, and in particular to Bogdan and Angi.

Contents

1 Introduction **1**

2 Background **7**

2.1 Opportunistic Networking 7

2.2 Bluetooth Discovering Procedure 9

2.3 Neighbour Discovery in Bluetooth 11

3 Fast and Energy-Efficient Neighbour Discovery Scheme **13**

3.1 Neighbour Discovery Scheme 14

3.2 Parameter Space Search 16

3.3 Results 23

4 Discovery Mode Selection **31**

4.1 Overview 31

4.2 Evaluation 34

4.3 Implementation 37

4.4 Results 39

5 Related Work **44**

6 Conclusions **48**

Bibliography **50**

List of Tables

3.1 Simulation precision vs. number of iterations 21

3.2 Parameter space explored in simulation 23

3.3 Parameter values for points with minimum mean discovery times 26

3.4 Power consumption of the Cellink BTA-3100 device 29

3.5 Parameter values for good trade-off points (simulation) 30

4.1 Parameters for discovery modes used in the simulation 35

4.2 Mean discovery times between two devices in different modes [s] 35

4.3 Simulation parameters 37

4.4 Energy per successful contact 40

List of Figures

2.1 Bluetooth discovery process 10

3.1 Screenshot captured from the oscilloscope 17

3.2 3D graph of the mean discovery time using four fixed parameters 24

3.3 Mean discovery time vs relative weight of the two phases 25

3.4 Cumulative distribution functions of the discovery time 27

3.5 Mean discovery time vs. average baseband slots usage 29

3.6 Mean discovery time vs. power consumption of the discovery process . . 30

4.1 Discovery modes used in the simulation 34

4.2 Discovery times times are exponentially distributed 36

4.3 Recent-activity level scheme 38

4.4 Location-based scheme 39

4.5 Performance of the five schemes versus number of nodes 40

4.6 Performance of the five schemes versus number of attraction areas 41

4.7 Performance of the adaptive, recent-activity scheme versus $t_{nocontact}$ . . . 42

4.8 Performance of the adaptive, location-based scheme versus *cell size* . . . 42

4.9 Throughput of the five schemes 43

Chapter 1

Introduction

This thesis focuses on optimizing the time for communication establishment and the power consumption of the neighbour discovery process in Bluetooth-based mobile *ad-hoc opportunistic networks*. We introduce a novel neighbour discovery scheme for Bluetooth and evaluate its performance for a large set of parameters. We further propose and evaluate two adaptive algorithms for dynamically selecting the parameters of the neighbour discovery process in order to switch between low-power. slow discovery and high-power. fast discovery modes. depending on context.

The idea of *opportunistic networking* [28. 7. 13. 17] is that. in the absence of global connectivity. content may be forwarded between mobile devices by taking advantage of communication opportunities that arise in the course of user mobility. The typical scenario in this kind of networks is that of two users carrying mobile devices (e.g.. PDAs. smartphones) with wireless networking capabilities (e.g.. Bluetooth. Wi-Fi). walking past each other and exchanging data (e.g.. news articles. weather reports. multimedia files) during the period of time they are in-range.

Bluetooth-enabled mobile devices are particularly suitable for this kind of new communication applications because they offer low-power short-range data transfers [10]. In addition. since such devices become increasingly pervasive. almost anyone with a Bluetooth device in her/his pocket becomes a potential participant in the forwarding process.

To enable the above scenario with Bluetooth, a scheme is required for fast establishment of communication between two devices when they become neighbours. In order to initiate communication, neighbouring devices first need to discover each other, i.e., learn about each other's presence. This implies that a continuous process for scanning the surroundings i.e., the *neighbour discovery process*, must be run on each device.

A key concern in mobile ad-hoc networks is power consumption: due to the limited battery life of mobile devices, each device needs to conserve energy. Previous user studies [28] have shown that high battery consumption on the devices limits the wide deployment of such ad-hoc Bluetooth networks: while the Bluetooth data transfer is low-power, the discovery process may consume considerable amounts of energy, due to its continuous nature.

The Bluetooth standard itself is ill-suited for this unanticipated peer-to-peer use due to: i) its *asymmetric* master-slave discovery design in which one device is actively sending search beacons (inquiry mode) and another device is passively waiting for beacons (scan mode), and ii) other artifacts such as the necessity for extensive frequency searching for a match between devices. Opportunistic communications call for ad-hoc Bluetooth networks in which devices use a *symmetric* neighbor discovery process: each device continuously alternates between the active and passive roles. To maximize the chances that random encounters will produce the desired opportunistic data transfer, the devices need to be in neighbour discovery mode either continuously or at least periodically.

Previous studies [22, 4, 30] have considered neighbour discovery in Bluetooth version 1.1 [2]. The best experimentally validated discovery scheme reports a minimum discovery time of 8 seconds [4]. In terms of power consumption, with this scheme, a device is engaged in baseband activity (transmit/receive) 51% of the time, on average. These results make opportunistic networking with Bluetooth unfeasible for two reasons:

1. The probability of establishing contact in the random passer-by setting is low,

independent of the frequency of the neighbor search process. Indeed, assuming that two people walk past each other at a leisurely speed of 1 m/s and considering the regular Bluetooth range of 10 m, the two devices will be in-range for roughly 10 seconds. Since 8 seconds are necessary just for the devices to sense each other and establish communication, the window of opportunity for information exchange will likely be missed in a lot of cases.

2. Due to the inherent behavior of the symmetric process in neighbor discovery on common devices, a user involved in such Bluetooth networks drains her/his device's battery sooner than a typical work day [28].

In the first part of this thesis, we propose a fast and energy efficient neighbor discovery scheme for opportunistic networks with Bluetooth. By exhaustively varying all possible parameters in the parameter space of this discovery scheme, we find a solution that dramatically improves *both* discovery time *and* power consumption over their best known counterparts. Our main observation is that during the inquiry phase, a device is continuously engaged in baseband activities (i.e., transmitting search beacons or receiving responses to beacons), thus consuming much more power than in the scanning phase when the device is only active for a typically short "listening" window. Hence, the key to finding the best of the two worlds i.e., discovery speed and power consumption, is to independently vary the mean times spent by the device in each of the two phases and the size of the "listening" window. This enables us to find parameter settings offering a very short mean discovery time and reduced power consumption.

In order to efficiently search the multi-dimensional parameter space, we have developed a simulator closely matching the implementation of real Bluetooth devices. The simulator enables us to search a much larger parameter space than it would be possible by measurement. To validate our simulations, we compare our simulation results with measurements. In particular, we confirm the discovery time and power consumption for the best parameter setting(s) found through simulation on a real experimental platform composed of two Cellink BTA-3100 devices with a Silicon Wave chipset and compatible with v1.2 of the Bluetooth protocol. Another novel contribution of our work is our methodology for carefully setting up simulation and measurements to avoid synchronization artifacts. We introduce a systematic methodology for experiment randomization to set the instant at which two devices come into range in a statistically meaningful way.

The results of this first part can be summarized as follows:

- Compared to the previous study of the neighbour discovery process based on Bluetooth 1.1 presented by Bohman *et al.* [4], we achieve a two-fold improvement in the minimum discovery time and an estimated two-fold improvement in power consumption (this metric was not considered in this previous work).

- When we consider our full range of improvements and the Bluetooth 1.2 protocol changes, we obtain a minimum mean neighbor discovery time of 0.2 seconds.

- For a reasonable average discovery interval of 1 second, power consumption of our neighbor discovery process is as low as one tenth of the power consumption of transmitting, or alternatively, as low as 1.5 times the power consumption of the device while in the idle mode.

Based on the results of the parameter space search, we learn that there is a tradeoff between the speed of neighbour discovery and the power consumption of the device. Depending on the choice of parameters, there is a range of discovery modes we might select from low-power, slow discovery to high-power, fast discovery. For a certain device, the best choice might depend on many factors including user preference and remaining battery life. While there is no single optimal discovery mode, ideally, we would like to maximize battery life while minimizing the lost opportunities for communication (due to slow discovery).

In the second part of this thesis, we propose two adaptive schemes for selecting the discovery mode based on locally-available information. Both schemes are based on the observation from real life, that wherever another person is encountered, there will likely be other contacts to follow shortly. Such is the case, for example, in classrooms, shopping complexes, public transport. The first scheme use the level of recent activity as an indicator for selecting the discovery mode. The second scheme assumes access to a positioning system and remembers previous contacts and their location. It then uses these patterns to predict future activity.

To evaluate our adaptive neighbour discovery selection schemes, we use a mobility simulator implementing a modified Random Waypoint [5] model to evaluate the performance of these two adaptive schemes along with some naïve, static discovery mode schemes. Our results show promising performance for the two adaptive schemes when compared to the naïve schemes: i) 50% less energy spent per contact and ii) up to 8% better throughput. When compared to the highest power mode available, although less contacts are made, the power consumption is far superior. While this investigation has its limitations, due to being unable to fully mirror the periodical patterns of movement of people in the real world, we believe both schemes are good candidates for implementation in a real-world setting.

To our knowledge, this work is the first one to consider the power consumption as a factor in the search for a solution to the neighbour discovery problem. Our work is also the first to show the feasibility of opportunistic networking on currently available Bluetooth hardware.

The rest of this dissertation is organized as follows. Chapter 2 provides the background on the discovery process in Bluetooth. Chapter 3 presents our neighbour discovery scheme and the experimental evaluation of its performance as a function of its parameters. In chapter 4, we present our two schemes for dynamically selecting the discovery mode and their performance evaluation. We discuss the related work in chapter 5 and conclude in chapter 6.

Chapter 2

Background

In this chapter, we introduce the concepts and terminology behind the work presented in this dissertation. First, to motivate our work, we present the concept of opportunistic networking and the challenges of the neighbour discovery process. Second, we present the discovering procedure in Bluetooth. At last, we talk about how the neighbour discovery process used in opportunistic networking would be implemented in Bluetooth.

2.1 Opportunistic Networking

With the advent of mobile devices, such as, laptops, PDAs and mobile phones, with short-range wireless networking capabilities, such as, Bluetooth, new types of networking applications are possible even in the absence of global connectivity. The mobility of the users that carry the devices can be exploited for data transport even when there is no end-to-end path between the source and destination nodes. A number of recent proposals [7, 28] investigate the idea of *opportunistic networking* or, alternatively, *pocket switched networking* [13]: propagating data in an epidemic fashion through 'opportunistic' data exchanges that occur when mobile devices come into wireless range due to the mobility of their users.

An example where this type of networking might be useful are remote and rural areas which have sporadic access to broadband infrastructure [14]. Another potential use would be networking when access infrastructure has failed, such as under disaster conditions. Another study [28] suggests that within particular user communities e.g., students on a university campus, targeted content might be forwarded using user mobility as a network transport mechanism. Finally, one can easily imagine generic data dissemination through epidemic propagation involving data of common interest such as MP3 music, news articles, weather reports or browsed web content. Such a system would work in the same way as peer-to-peer file sharing systems, where users agree to donate computational resources to the community, in return for access to the shared content.

The typical scenario in opportunistic networking involves two users that carry mobile devices exchanging data as they walk past each other. A key step in establishing contact between these users is the *neighbour discovery process*: nodes need to learn about each other's presence before establishing a connection. The absence of any infrastructure means that devices have to continuously scan their surroundings to detect nearby nodes.

The challenges presented by this continuous discovery process are two-fold:

Short discovery times: Communication opportunities within such networks, are usually brief, on the order of a few seconds. Two users carrying devices that have a radio range of 10m, such as that of Bluetooth, and walking towards each other at a normal speed of 2m/s will have a window of opportunity of only 5s during which to discover each other's presence and establish communication. Hence, the discovery process needs to be *fast* in order to enable devices to take advantage of random encounters.

Low power consumption: While most subsystems on mobile devices have seen a great deal of progress in recent years, power remains a scarce resource and must be conserved as much as possible. The discovery process, being a continuous process run by the device, needs to be *energy-efficient* in order to avoid prematurely draining the battery.

While it is desirable to have both characteristics at the same time, fast discovery and low power consumption, it is clear that at some level a trade-off between the two needs to be achieved. A device might use more energy-intensive discovery modes when quick discovery times are essential, and conversely, low-power modes when conserving battery is more important.

2.2 Bluetooth Discovering Procedure

Bluetooth [3] is a prime candidate technology for opportunistic networking due to its low-power consumption and its wide deployment on mobile devices.

Typical Bluetooth devices use an asymmetric protocol for discovering each other. One of the devices performs the active role, sending beacons and listening to replies. This device is known as the *inquiring* device, or otherwise, performing the *inquiry* procedure. The other device, in the passive role, performs the *inquiry scan* procedure, which consists in listening to beacons and sending responses.

Figure 2.1 illustrates the main aspects of the discovery process. The Bluetooth physical layer is based on a frequency hopping scheme, in which devices use one of the 79 available frequencies according to a pseudo-random scheme. For discovery, a device uses a special hopping sequence that covers only 32 out of the 79 frequencies. The inquiring device sends 1600 beacons per second cycling very fast through the available frequencies. The 32 inquiry frequencies are split into two *trains* of 16 frequencies each, called A and B (see figure 2.1). The inquiring device uses the frequencies only in one of the trains at a time. It switches trains every 2.56 seconds. The scanning device periodically listens to beacons during a usually short window of time. Two Bluetooth parameters, called the *scan interval* and the *scan window*, control the frequency and the duration of the listening windows. If the device receives a beacon during a scan window, it waits for 625 μs and

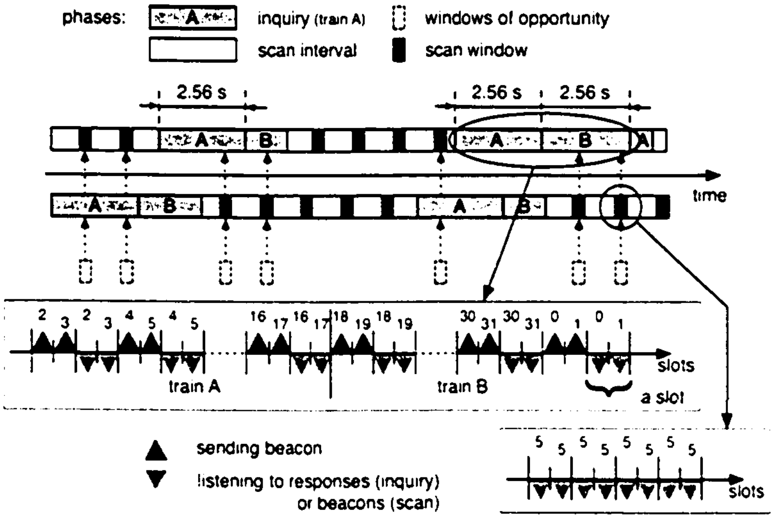


Figure 2.1: Bluetooth discovery process

sends a response on the frequency of the beacon. The discovery process completes when the inquiring device receives the response. The scanning device follows a much slower hopping pattern changing the frequency it listens to only every 1.28 s.

The physical channel used by Bluetooth devices is divided into *time slots* of 625 μs. During the discovery procedure, the inquiring device sends two beacons in each slot on two different frequencies and listens to responses during the next slot on the frequencies of the beacons. Each Bluetooth device has an internal clock that determines the timing and frequency hopping of the transceiver. The clock is implemented as a 28-bit counter whose least significant bit ticks in units of 312.5 μs. It solely determines the frequency to use in the inquiry or inquiry scanning procedures according to the equations below:

$$\begin{aligned} F_{inquiry} &= [CLK_{16-12} + k_{\text{offset}} + (CLK_{4-2,0} \\ &\qquad -CLK_{16-12}) \bmod 16] \bmod 32 \\ F_{scan} &= [CLK_{16-12} + N] \bmod 32 \qquad (2.1) \end{aligned}$$

where k_{offset} is 24 for train A and 8 for train B, respectively. N is a counter incremented after each response to a beacon. By CLK_{16-12}, we denote the 5-bit sequence from bit 16 to 12 of the clock.

As it can be seen from these formulae, the scanning frequency changes whenever bit 12 of the clock changes, that is, every 1.28 seconds, while the inquiry frequency changes with every tick of the clock, that is, every 312.5 μs.

There are two differences between the discovery procedure in Bluetooth version 1.1 and 1.2. The first one is the random backoff defined in v1.1 and dropped in v1.2: after receiving a first beacon, the scanning device would go into a backoff period of up to 0.64 s and only after receiving a second beacon from the same device, it would send a response. It has been shown that the backoff period potentially doubles discovery times at the slight benefit of avoiding collisions, which are anyway fairly unlikely [30]. The second difference is the *interlaced* inquiry scanning mode added in v1.2. If a device performs interlaced scanning, as opposed to the standard scanning procedure described above, then each (standard) scan window on frequency F_{scan} is followed immediately by a scan window (of the same length) on frequency $(F_{scan} + 16) \bmod 32$. Due to the way the inquiry train membership evolves over time, it is guaranteed that the two scanning frequencies will be in different trains for any given inquiring device.

2.3 Neighbour Discovery in Bluetooth

In an ad-hoc scenario required for opportunistic communications, the roles of Bluetooth devices (active versus passive) cannot be predefined, because if both devices had the same role, they would never discover each other. Instead, devices need to alternate between the inquiry (active) and inquiry scan (passive) modes, as pointed out in previous studies [22], which have considered neighbour discovery with Bluetooth v1.1. The *residence time* is the time a device performs a given role (active or passive). Salonidis *et al.* [22] have shown that the residence times should be random in order to ensure bounded discovery times. Otherwise, if residence times were deterministic, two devices with synchronized schedules for alternating roles, would never discover each other.

Bohman *et al.* [4] have further studied the Bluetooth neighbor discovery scheme by means of simulation and measurements on an experimental testbed. They have defined the residence time, as composed of a fixed part and a variable (random) part, that is distributed either uniformly or exponentially. By trying different parameters for the residence time, they found out that when the mean residence time in each role is around 2.5 seconds, the *mean discovery time*, i.e., the average time from the moment two devices are in range until one of them discovers the other, is around 8 seconds. They also concluded that there is little difference between using a uniform or exponential distribution for the variable part of the residence time.

Chapter 3

Fast and Energy-Efficient Neighbour Discovery Scheme

In this chapter, we present our neighbour discovery scheme for Bluetooth and evaluate its performance, in terms of discovery time and power consumption. In section 3.1, we describe the scheme and discuss its parameters. The performance of this scheme depends on the values of its parameters. We are interested in finding the optimal values for these parameters. In section 3.2, we explain how we perform the search for the optimal parameters through simulation, and how we validate the simulation through measurements on real devices. Finally, in section 3.3, we present the results of the search process, including those parameter values that yield the best trade-off between power consumption and mean discovery time.

3.1 Neighbour Discovery Scheme

In this section, we present our neighbour discovery scheme. We extend the scheme of Bohman *et al.* [4] to use unequal mean residence times. At the same time we vary the timing and frequency of scan windows and we investigate the new interlaced scanning mode of Bluetooth v1.2 [3]. Our scheme does not require any modification of the standard and can be implemented on currently available hardware.

While performing the active role (inquiring) a device is continuously engaged in baseband activities, sending beacons and listening for responses. During the passive role (scanning) the device is usually listening for beacons for a fraction of time (the scan window). The power consumption is thus, usually, much greater during the active role. Starting from this observation, for an energy-efficient scheme, we propose to differentiate the durations of the active and passive phases, in contrast with previous work [22, 4] which only considered equal mean residence times for the two roles.

The Bluetooth standard [3] allows the user to specify the *scan_window* and *scan_interval* parameters: how often and for how long the passive device is listening while in the scan phase. However, previous work [22, 4] only considered the default values of these parameters: scan windows that last 11.25 ms and are scheduled every 1.28 s. We propose to vary the duration and frequency of the scan windows.

By varying the relative weight of the two phases and the timing of the scan windows we are looking for two-fold benefits: a reduction in power consumption and an improvement of the mean discovery time. Intuitively, spending more time in the passive phase should give us the sought for power consumption reduction, while increasing the frequency and/or duration of scan windows should increase the chances of being discovered, and thus, compensate for spending less time in the active role.

In addition, unlike previous work [22, 4] that was based on Bluetooth v1.1 [2], we explore the use of the new *interlaced scanning mode* of Bluetooth v1.2 in addition to the standard one.

The scheme

We define the residence times for the two phases of the discovery process as composed of a fixed (constant) part and a variable (random) part, as shown in the following equations, where rand(x, y) denotes an integer-valued uniformly distributed random variable with values in the interval (x, y):

$$\begin{aligned} T_{inquiry} &= C_{inq} + \text{rand}(0, 2V_{inq}) \\ T_{scan} &= C_{scan} + \text{rand}(0, 2V_{scan}) \end{aligned} \tag{3.1}$$

Note that the mean values of the variable parts are thus V_{inq} and V_{scan}.

Parameter space

The discovery process, as described above, is governed by seven parameters:

- C_{inq}, V_{inq}, C_{scan}, V_{scan}, the parameters of the random variables that determine the residence time in the inquiry (active) and, respectively, in the scan (passive) phase.
- *scan_window* and *scan_interval*, determine the frequency and duration of the scan windows. The *scan_window* and *scan_interval* can take values in the interval $[11.25, 2560]$ms.
- the inquiry scanning mode: *interlaced* vs. *standard*.

These seven parameters define the *parameter space* of the discovery process. We call a particular set of values for these parameters, a *point* in the parameter space. We are interested in exploring the performance of the discovery scheme at a large set of points. The metrics of interest are the *mean discovery time* and the *mean power consumption*. The discovery time is defined as the time elapsed between the instant the devices are in the communication range until one of the devices discovers the other. The power consumption is derived (rather than measured) from the parameters of the discovery process.

3.2 Parameter Space Search

This section presents our methodology for evaluating the performance of the proposed neighbour discovery scheme. First, we provide an overview of the experimental evaluation process. Next, we describe our experimental platform and how we modeled its characteristics in the simulator. Finally, we present our methodology for experiment randomization so as to set the instants at which two devices come into range in a statistically meaningful way.

3.2.1 Overview

Since the parameter space is seven-dimensional, it would be unfeasible to explore it through measurements on real devices. We have implemented a discrete-event based simulator that closely models the discovery process as specified in the Bluetooth specification [3] with the specific features of the Bluetooth devices that we used in our measurement, as explained in detail in the next section. We use this simulator to explore a large number of points in the parameter space and we validate the values obtained from the simulator through empirical measurements for some selected points.

Most of the previous work has studied discovery schemes either analytically or by means of simulation. In contrast, we have paid considerable attention to validation and comparisons with measurements on an experimental platform. When performing experiments with Bluetooth devices, we have noticed that performance results depend heavily on the way an experiment is set up. We consider a scenario of two devices entering the radio range of each other. This means that they are able to communicate at random time instants. If we want to estimate mean discovery times, the estimation needs to rely on correctly randomized start times of experiments and the clock offset of devices so as to match the scenario. This is required to eliminate synchronization artifacts between experiments that may lead to wrong results.

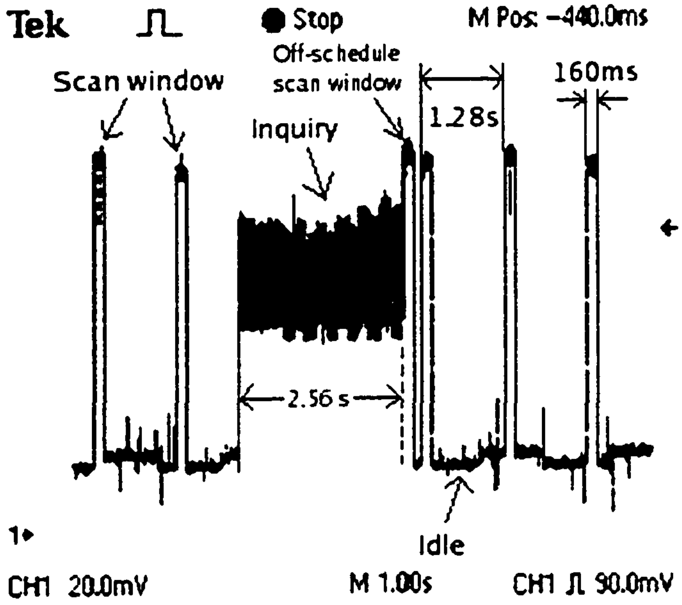


Figure 3.1: Screenshot captured from the oscilloscope

3.2.2 Experimental Platform

In this section, we explain how we used an oscilloscope to identify the design decisions taken by the implementors of the Bluetooth firmware in the devices we used in our experiments. We modeled these decisions appropriately in our simulator.

We use two Cellink BTA-3100 devices with a Silicon Wave chipset and compatible with v1.2 of the Bluetooth protocol. Some implementation details of the Bluetooth protocol are left up to the manufacturer, for example, the scheduling of the scan windows following an inquiry period. Another example of an implementation-dependent parameter is the time between inquiry train changes, for which only a lower bound of 2.56 s is specified. We had to identify these design decisions and model them accordingly in our simulator. Because we did not have access to the firmware of the devices, we used a novel technique based on measurements taken with an oscilloscope to investigate the manufacturer's design decisions for our devices. It turns out that the instantaneous power consumption patterns can be associated very clearly with ongoing baseband activities (idle mode, scan windows, or inquiry periods) as can be seen in Figure 3.1. In order to measure the instantaneous power consumption we have placed a precision resistor in series with the Bluetooth device and we have measured the voltage drop across this resistor with the aid of a digital storage oscilloscope. Using this method we identified the following device-specific characteristics:

- Scan windows always start on the same schedule at clock values equal to $0.64s + k * scan\_interval$ unless there is already an ongoing inquiry, in which case the window is skipped. At the end of an inquiry, if a scan window was skipped, an out-of-schedule scan window (see Figure 3.1) starts in the slot immediately following the inquiry. The next scan windows are scheduled regularly.

- Each inquiry starts with the same train. The train changes every 2.56 s. According to the Bluetooth standard, the length of an inquiry is specified in units of 1.28 s. However, since an *inquiry cancel* command takes effect immediately, it is possible to do inquiry for periods of time with a finer granularity. If at the instant the inquiry command is issued, there is an ongoing scan window, then the inquiry start is delayed until the scan window finishes.

- Increasing N in the scanning frequency equation (Eq. 2.1) after each response, is not implemented in our devices.

- Turning off inquiry scanning makes the device undiscoverable. We use this feature to emulate the entry of a device into the range of another one. We have observed that turning inquiry scanning on and off takes effect immediately. After inquiry scanning is restarted, the same schedule for the scan windows is preserved.

- Scan windows larger than 512 slots (0.32 s) are not well supported by our devices. For this reason, we used 512 slots as an upper bound for the length of the scan windows. This value is sufficient for all practical purposes.

Although these specific characteristics do not limit the generality of the results we present, in order to validate empirically the simulation results, we had to model these design decisions as accurately as possible in our simulator.

3.2.3 Randomizing experiments

In this section, we explain how we randomized our experiments in order to accurately model the scenario of interest. Randomization is a key part of setting up unbiased experiments.

The inquiry or scan frequency for a device is determined solely by its clock. More specifically, it is only the least significant 17 bits, denoted by CLK_{16-0} that matter (cf. Eq. 2.1). At the start of a measurement session or a simulation run, it should be equally likely for CLK_{16-0} of each device to have any value in the range.

For any device at a given instant, the inquiry trains consist of (possibly wrapping) blocks of 16 consecutive frequencies in the interval $[0, 31]$. Every 1.28 s, the train membership changes by sliding one unit to the "right"; so if initially train A is $[0, 15]$, after 1.28 s it will become $[1, 16]$, after another 1.28 s it will be $[2, 17]$, and so on. At the same time, the scan frequency increases (with wraparound) by one unit every 1.28 s. This implies that for two devices whose relative clock offset remains constant, the scan frequency of one device will always be in either train A or train B (depending on the offset) of the other device.

For the devices in our platform, inquiry always starts with the same train (let us assume it is A). If the scan frequency of one of the devices is always in train B of the other device, than this latter device will always need to inquire for at least 2.56 s (to allow for switching to train B) in order to discover the former device. This problem can become even worse, because our devices do not use the term X in Eq. 2.1 (if implemented, it would allow the scan frequency to "catch from behind" the other train after a certain number of responses).

Clearly, the relative clock offset of the two devices should be independent between experiment iterations and uniformly distributed over its range to guarantee unbiased experiments. Hence, we enforce this condition both in the simulator and in the way we set up our measurement sessions.

The Bluetooth specification does not provide a command for setting the clock value of a device. Hence, we have achieved randomization by power-cycling one of the devices after each experiment iteration. Since the experimental setup needs to be automatic, we have controlled power-cycling by connecting the devices to a USB external hub with per-port power switching support: whenever a device is power-cycled, its clock is initialized to zero. In order to randomize the relative clock offset between devices, we sleep for a random period of time between experiment iterations. The random sleep value is uniformly distributed between 0 and 40.96 s (which is the amount of time necessary for CLK_{16-0} to run over its range). By setting the experiments this way, the relative clock offset can have any value in its range with equal probability.

There is also another subtle issue in setting up an unbiased experiment. At the instant the devices get into the range of each other, each device could be in the inquiry or scan phases with the respective probabilities, and furthermore, they could be at any point during those phases. Thus, when we start the experiment, we cannot just randomly choose one of the phases and have the device start that phase from its beginning. In particular, for the "inquiry phase", since train changes occur every 2.56 s relative to the start of the inquiry, performing a short inquiry to emulate what is "left over" of an inquiry phase, does not suffice. Rather, we start the full inquiry phase and we emulate being out of range of the other device, until the moment of starting the experiment.

To emulate the instant at which devices come into the range of each other, we have used the Bluetooth command that allows turning inquiry scanning on and off. When the devices are supposed to be out of the range, we make both devices non-discoverable (inquiry scanning turned off). We turn inquiry scanning back on for both devices at

Number of iterations	Mean (sec)		Std. dev. (sec)	
	min	max	min	max
100	3.182	5.864	3.277	7.570
1,000	3.992	4.743	4.439	5.680
10,000	4.251	4.507	4.779	5.237
100,000	4.317	4.423	4.947	5.086
1,000,000	4.359	4.389	4.990	5.034

Table 3.1: Simulation precision vs. number of iterations

the emulated moment of getting into range (which is also the instant at which we start measuring the discovery time).

We had to remove another synchronization artifact: the scan windows are always on the same schedule relative to the power-on time of a device as described above. In order to randomize the start of experiments and the scheduling of the scan windows, we sleep for another random period of time uniformly distributed between 0 and the length of a scan window, before each experiment.

Experiment iterations

The randomization methodology results in large times for running experiment iterations with real devices (around 35-40 s per iteration for a typical point in the parameter space). Due to time constraints, we have chosen to run only 1000 iterations in our measurements with real devices (this takes approximately 11-12 hours to complete) for each points in the parameter space that we validate. The drawback is lower precision in the measured statistics.

To quantify the precision loss, we ran for a particular point in the parameter space, our simulator with various number of iterations. For each number of iterations, the experiment was repeated 100 times with different random seeds. Table 3.1 summarizes the results. For example, at 10,000 iterations, we see that depending on the random seed, the estimation error of the mean discovery time is up to 8%. As the number of iterations increases, the simulation converges to the "real" mean regardless of random seed. In our simulation, we chose to perform 10,000 iterations for each point in the parameter space, due to limited computational power.

Parameter	Values								
C_{inq}[s]	0.25	0.5	1.0	2.0	3.0	4.0	6.0	8.0	
V_{inq}[s]	0.25	0.75	1.5	2.0	2.5	3.0	4.0	4.5	5.0
C_{scan}[s]	0.25	0.5	1.0	2.0	3.0	4.0	6.0	8.0	
V_{scan}[s]	0.25	0.75	1.5	2.0	2.5	3.0	4.0	4.5	5.0
scan_window[ms]	11.25	22.50	45.00	90.00	180.00	312.50			
scan_interval[ms]	160.00	320.00	472.50	640.00	960.00	1280.00	2557.50		
scan_mode	standard	interlaced							

	Number of points explored
Standard scan	207.360
Interlaced scan	186.624
Total	352.152

Table 3.2: Parameter space explored in simulation

3.3 Results

In this section, we present our simulation results in terms of our two metrics of interest, the *mean discovery time* and the *mean power consumption*.

3.3.1 Overview

Table 3.2 summarizes the parameter subspace that we explored in the simulation by showing the values taken for each parameter and the total number of points. Note that *scan_window* should be always smaller than *scan_interval* so some combinations of values can be eliminated. In the interlaced scan mode, the actual window length is twice the *scan_window*, so only points that have 2 * *scan_window* < *scan_interval* are to be considered (which accounts for less points explored in the interlaced scanning mode, cf. table 3.2).

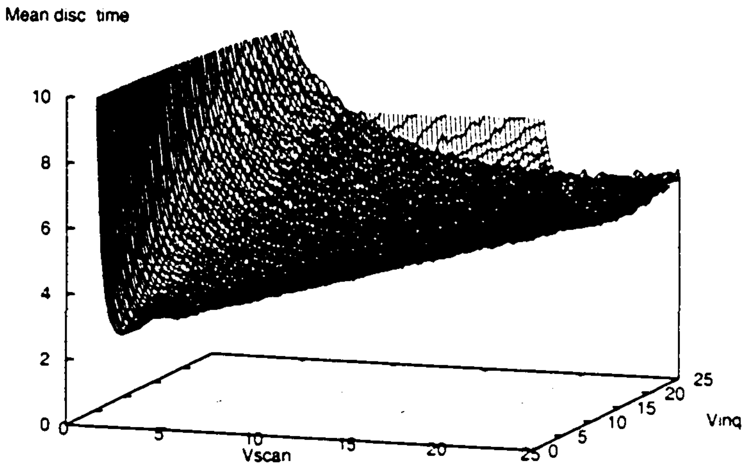


Figure 3.2: 3D graph of the mean discovery time using four fixed parameters

Presenting performance of our fast neighbor discovery scheme as a function of all seven parameters is not workable. So, in order to give the reader some feeling about the influence of parameters on the discovery process, we have fixed four of the parameters ($C_{inq} = C_{scan} = 0.25s$, $scan\_window = 0.64s$, $scan\_interval = 22.50ms$) and varied the remaining two. The resulting graph is shown in Figure 3.2. For this particular subspace, it appears that shorter discovery times are obtained when the residence times for the two phases are roughly equal.

Also, to get some intuition about the effect of using unequal mean residence times on the energy consumption and mean discovery time we fixed the scan window parameters ($scan\_window = 1.28s$, $scan\_interval = 11.25ms$), and we considered the ratio of the mean residence time in the inquiry and scan phases plotted versus the mean discovery time. Figure 3.3 shows the resulting graph with all points in the explored parameter subspace.

The scheme gives shorter discovery times when time spent in the two phases is roughly

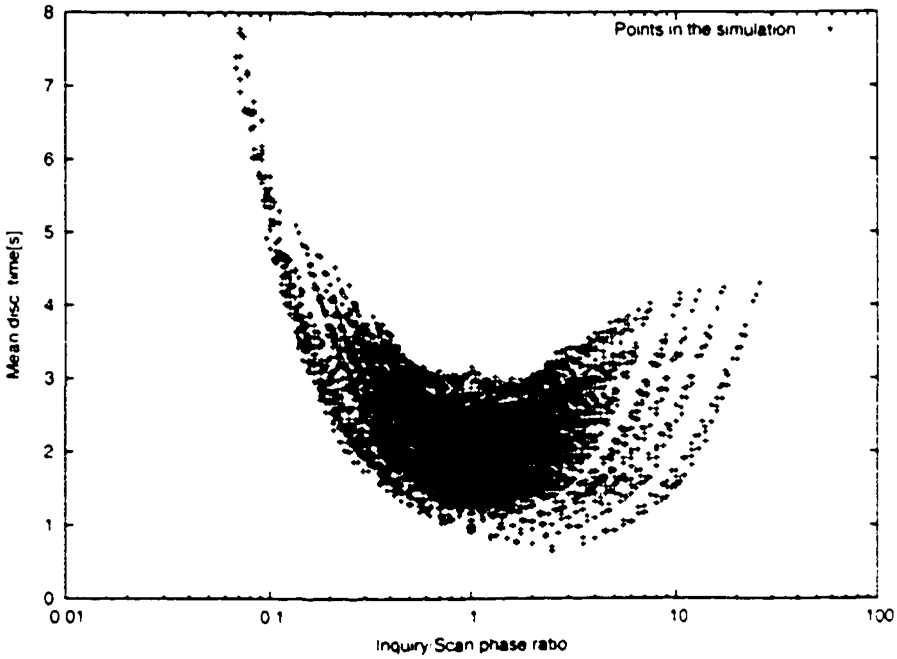


Figure 3.3: Mean discovery time vs relative weight of the two phases

equal with some bias towards spending more time in the inquiry phase. Because power consumption in the scan phase is very small (listening to beacons during a small scan window out of an interval) compared to the inquiry phase (device continuously engaged in transmission activity), we are interested in spending more time in the scan phase, if we want to reduce power consumption. We can also notice that reasonably short discovery times, e.g. 2 s, are obtained even when the scan phase is on the average 5 times longer than the inquiry phase.

3.3.2 Discovery time

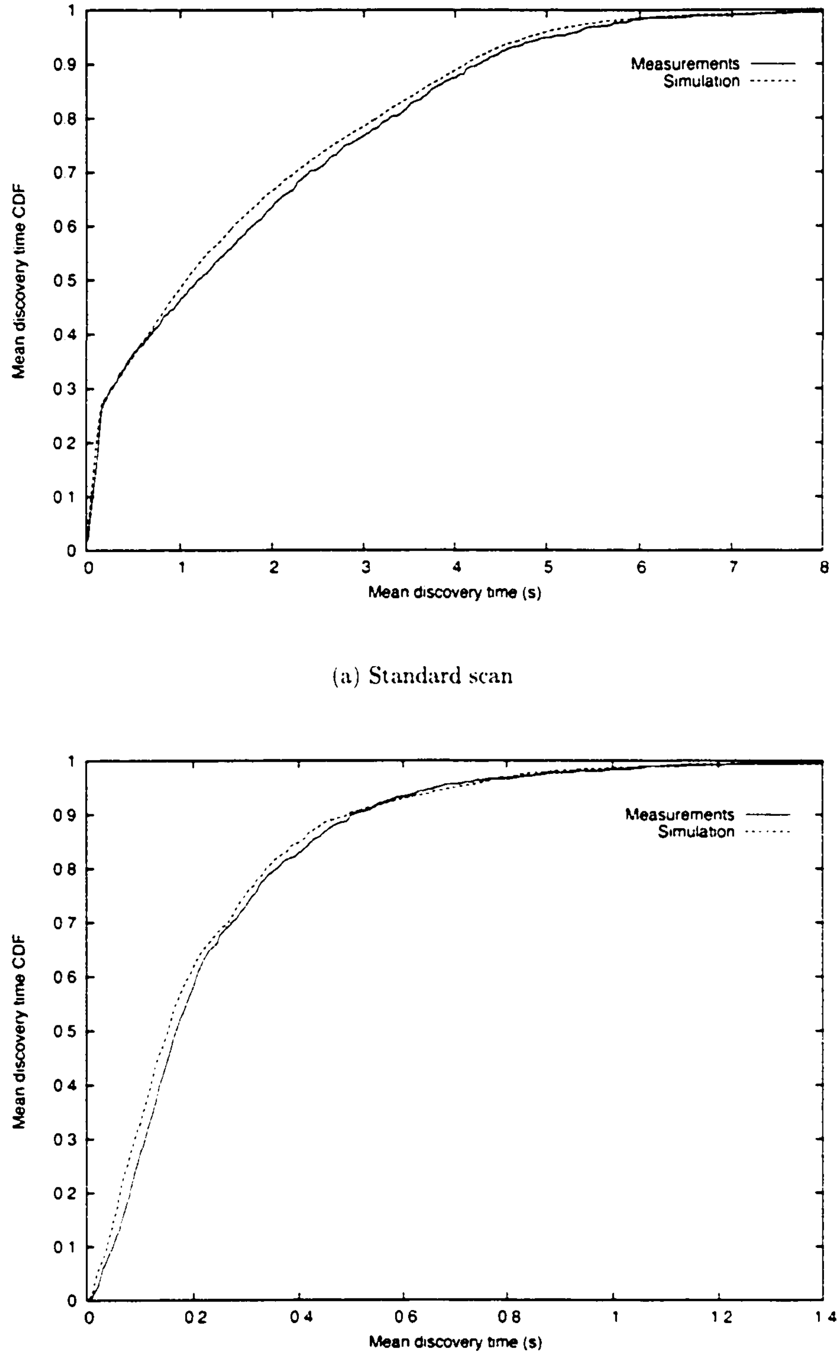
Table 3.3 shows the two points (for standard and interlaced inquiry scanning) that yield the minimum mean discovery times. For these points, we also ran experiments on real devices. Figure 3.4 shows the cumulative distribution functions of the discovery times obtained from simulation and measurements for the points yielding the minimum mean discovery times. We can see that the measurement results closely match those obtained

C_{inq}[s]	3.0		0.25	
V_{inq}[s]	0.25		0.25	
C_{scan}[s]	1		0.25	
V_{scan}[s]	1.5		0.25	
scan_window[ms]	90		45	
scan_interval[ms]	160		160	
scan_mode	standard		interlaced	
	Simul.	Meas.	Simul.	Meas.
Mean disc. time[s]	1.624	1.730	0.208	0.239
Std. dev.[s]	1.743	1.781	0.197	0.228
Median disc. time[s]	1.012	1.188	0.149	0.166

Table 3.3: Parameter values for points with minimum mean discovery times

through simulation. We also observe that the interlaced mode performs noticeably better due to its scanning of frequencies in both inquiry trains.

For comparison with the previous work that considered discovery in Bluetooth v1.1 [4], we have also examined a parameter subspace containing only the points considered in their simulations, that is, points for which $C_{inq} = C_{scan}$, $V_{inq} = V_{scan}$ and $scan\_interval$ = 1.28s, $scan\_window$ = 11.25ms. The minimum mean discovery time obtained for these points was 3.93 s. The improvement over the reported value of 8 s [4] is due to the removal of the random backoff procedure from Bluetooth v1.1. Our minimum over the whole parameter space using standard scanning shown in Table 3.3 represents a two-fold improvement of the mean discovery time. This results from using unequal mean residence times and different timing for the scan windows defined in our scheme.



(a) Standard scan

(b) Interlaced scan

Figure 3.4: Cumulative distribution functions of the discovery time

3.3.3 Power consumption

The second metric we consider is the power consumption of the discovery process. We derive power consumption from the parameters of the discovery process based on careful measurement of an operational prototype. Bluetooth devices divide time into slots used for various activities (inquiry, scanning, connections). There is a strong correlation between the number of slots that are dedicated to the discovery process and its power consumption. Moreover, less slots used for discovery means more slots available for other baseband activities such as serving ongoing connections. So, we relate power consumption to the percentage of baseband slots used in the discovery process out of the total number of slots. By power used by the discovery process, we refer only to power spent in excess of the idle mode consumption.

For each point in the simulation, we have derived a simple formula for the average percentage of slots used during discovery. For each value of the mean discovery time, we retain the point that has the minimum slot usage; the resulting graph is shown in Figure 3.5.

As expected, spending more power results in shorter discovery times. Interlaced scanning outperforms standard scanning over the whole range of interest. To compare with previous work, the scheme proposed by Bohman *et al.* [4] would use approximately 51% of slots for discovery, if we consider only equal mean residence times and default scan window parameters. If we take their lowest value of the mean discovery time (3.93 s), our scheme obtains the same value with a two-fold reduction in power consumption, because it only uses 25.5% of slots.

Although the slot usage gives a good estimate of the real power consumption, some devices use more power for one of the discovery states, inquiry or scanning, thus the real consumption might differ. We have measured the power consumption of Cellink devices in the idle mode and in the two discovery states as shown in Table 3.4 (see also Section 3.1 for the description of the measurements). We then use these values

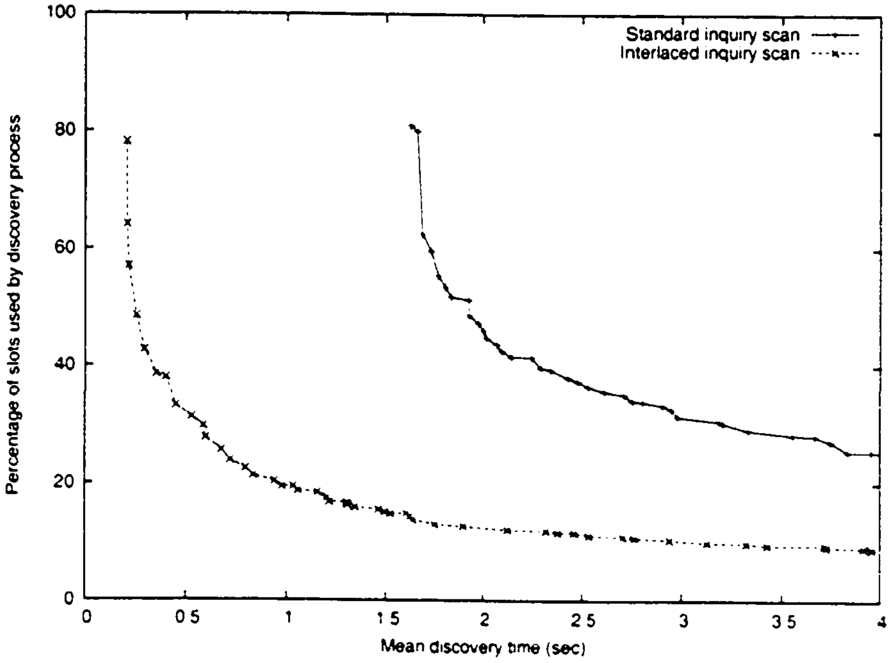


Figure 3.5: Mean discovery time vs. average baseband slots usage

State	Power consumption
Idle	91 mW
Inquiry	280 mW
Inquiry Scan	437 mW

Table 3.4: Power consumption of the Cellink BTA-3100 device

to redraw the power vs. discovery time graph with the power expressed in milliwatts (cf. Figure 3.6). The difference in graphs comes from the fact that the scan mode uses considerably more power than the inquiry mode on the devices in our experimentation platform. We observe from the power versus discovery time graph that if discovery times under 8 s are not essential, interlaced scanning offers power modes only slightly higher than the idle mode consumption in Bluetooth v1.1.

Table 3.5 gives parameter values for points that yield good trade-off between the low mean discovery times and power consumption (they are on the knee of the curve discovery time vs. power consumption, cf. Figure 3.6).

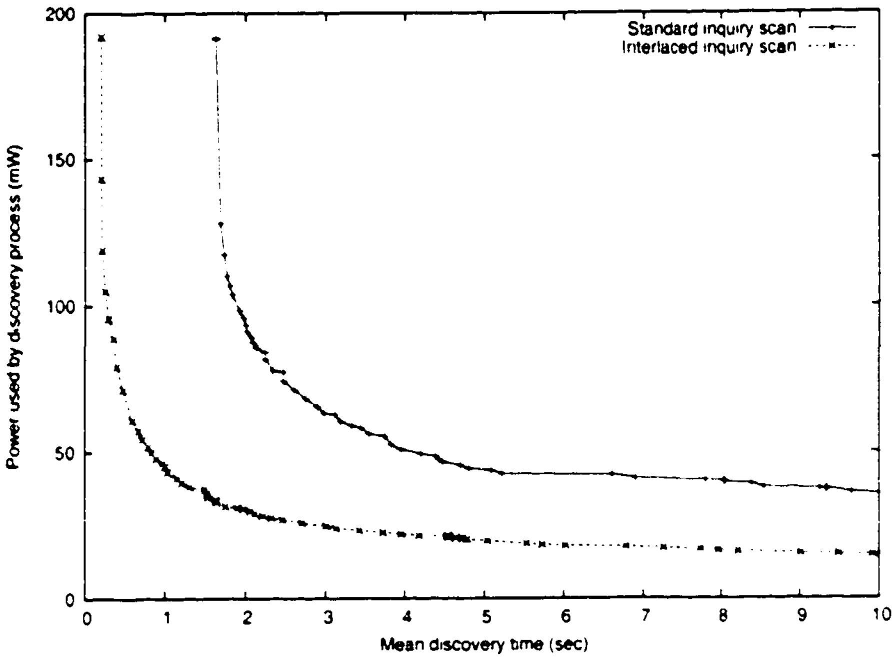


Figure 3.6: Mean discovery time vs. power consumption of the discovery process

C_{inq}[s]	3	0.25
V_{inq}[s]	0.25	0.25
C_{scan}[s]	3	0.25
V_{scan}[s]	5	2.5
scan_window[ms]	11.25	11.25
scan_interval[ms]	320	472
scan_mode	standard	interlaced
Mean disc. time[s]	2.98	1.03
Std. dev.[s]	2.90	0.96
Median disc. time[s]	2.21	0.74
Slot usage	31.4%	19.4%

Table 3.5: Parameter values for good trade-off points (simulation)

Chapter 4

Discovery Mode Selection

4.1 Overview

In section 3.3. we explored the performance of our neighbour discovery scheme for a large space of parameter values. We explored the performance of various *discovery modes* (i.e., sets of parameters for our neighbour discovery scheme). with wide ranging results. from discovery modes with very fast mean discovery times to very energy-efficient modes. The question that arises is what discovery mode should be used by a Bluetooth device in the envisaged opportunistic networking scenario.

From the point of view of the trade-off that they achieve between discovery time and power consumption. we could classify discovery modes as ranging from 'aggressive' (fast discovery. energy intensive) to 'lazy' (slow discovery. low power consumption).

It is clear that. in a generic fashion. no single discovery mode could represent the best choice. Aggressive modes would drain the battery of the mobile device too soon. while lazy modes would miss many communication opportunities. The choice of a discovery mode could be user-specified according to the priorities at a certain moment: short discovery times or low battery usage. Alternatively. devices could dynamically switch modes based on some global information such as knowledge about nearby devices. While such global information might be available, for example a cellular telephony network might provide the number of users in a certain cell, it is likely that privacy issues would make such a mechanism unfeasible.

Instead, we propose two schemes for dynamically switching discovery modes based solely on local knowledge. Ghosh *et al* [12] observe that in the real world, users tend to move in a non-random fashion: they spend a considerable amount at a few places and "orbit" between those places. Our two schemes are based on a variant of this observation: "wherever a device is encountered, it is likely that there are more devices". Next, we describe our adaptive schemes:

Recent activity level scheme In this scheme, a device switches to more aggressive modes whenever another device is discovered, and conversely, goes back to lazier modes when no device has been seen for a while. A typical scenario where the benefits of this scheme are visible, is that of a user entering a crowded place, such as a shopping complex or a subway station. After the first peer device is discovered, the user's mobile device is put into a more aggressive mode, thus discovering increasing its chances of discovering the remaining devices. When the user exits the place, the device is switched back into more economic modes of operation.

Location, past activity scheme In this scheme, a device has access to a global positioning system, and memorizes past contacts and their location. Whenever, a location where contacts have been made previously is approached, the device switches to a more aggressive mode. This scheme has a learning phase: initially it should perform poorly as it has no knowledge about a user's movement patterns and, as knowledge is gained, performance should improve.

We plan to use a mobile network simulator to investigate the performance of these dynamic schemes along with three simple static schemes, for comparison:

1. A scheme that constantly uses a lazy discovery mode. This scheme should be the reference in terms of energy efficiency.

2. A scheme that uses an aggressive discovery mode, thus being the reference in terms of fast discovery times.

3. A scheme that uses the previously best known mode from [4] (discovery modes that use equal residence times and default scan window timing).

In our evaluation of these schemes, we focus on two general aspects of performance:

Successful contacts ratio Ideally, all possible communication opportunities, i.e. two device are in range, are taken advantage of. In practice, however, if the discovery time is greater than the time devices are in range, they will not discover each other. Such an occurrence is a missed opportunity, as opposed to a successful contact, when devices do discover each other. Clearly, we would like to maximize the successful contact ratio out of the encountered opportunities.

Throughput Short discovery times, in addition to being desirable to avoid missing communication opportunities, have another benefit: they leave more time for actual data transfers to occur. Hence, discovering a neighbouring node faster, implies that the chances of a successful data transfer increase. We measure the throughput (i.e., the number of completed transfers) of our simulated ad hoc network for each of the five schemes.

Both metrics of performance must be weighed against the energy consumption (otherwise, clearly the most aggressive discovery mode outweighs all the others in performance). Hence, in our simulation, we measure average power consumption of the devices and present it alongside the performance results.

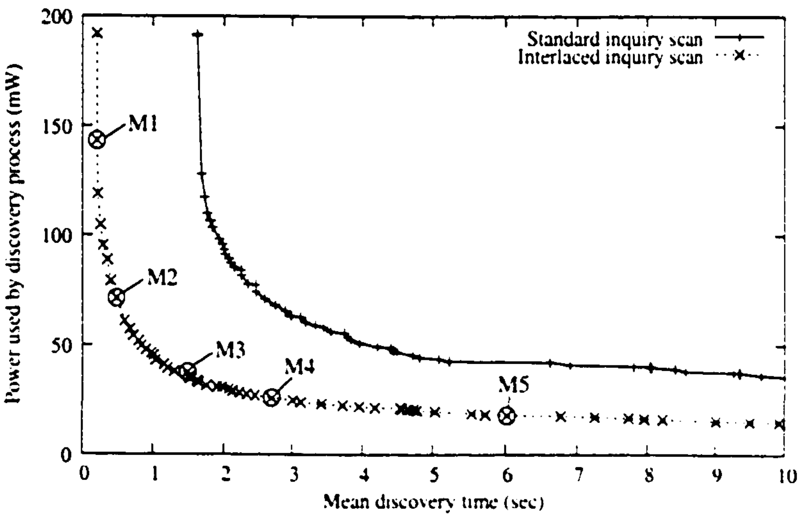


Figure 4.1: Discovery modes used in the simulation

4.2 Evaluation

In this section, we describe our evaluation methodology. First, we describe how we implement the discovery process in the simulation. Second, we describe the mobility model that we use and its parameters. Finally, we describe the message generator for our network.

Discovery modes

The devices whose features are emulated in our simulation are the same Cellink devices used in chapter 3. In particular, discovery times and power consumption, are based on the measurements done in the previous chapter.

We chose five discovery modes for use in our simulation, as shown in figure 4.1, where M_1 is the most aggressive mode and M_5 is the laziest. At any given time, a node in our simulation uses one of these discovery modes. Table 4.1 shows the parameters of these discovery modes and their corresponding average power consumption. For comparison with previous work, we also use the best mode given by the scheme of Bohman *et al* [4], which we call M_b.

	scan_interval[ms]	*scan_window*[ms]	C_{inq}[s]	V_{inq}[s]	C_{scan}[s]	V_{scan}[s]	Power[mW]
M_1	160	45	0.25	0.25	0.25	0.25	108
M_2	320	22.50	0.25	0.25	0.50	0.75	114
M_3	640	22.50	0.25	0.25	0.50	2.50	127
M_4	640	22.50	0.25	0.25	1.00	5.00	161
M_5	960	22.50	0.50	0.25	8.00	5.00	270
M_b	1024	11.25	1.75	2.00	1.75	2.00	173

Table 4.1: Parameters for discovery modes used in the simulation

	M_1	M_2	M_3	M_4	M_5
M_1	0.21	0.24	0.38	0.55	0.78
M_2	0.24	0.45	0.76	1.08	1.53
M_3	0.38	0.76	1.45	2.06	2.98
M_4	0.55	1.08	2.06	2.89	4.15
M_5	0.78	1.53	2.98	4.15	5.92

Table 4.2: Mean discovery times between two devices in different modes [s]

In chapter 3, we evaluated the mean discovery time for these discovery modes, when the same mode is in use on both devices. In order to realistically model the discovery process in our node mobility simulation, we had to consider the case where two nodes, that are in range, use different discovery modes. Hence, we modified the discrete-event simulator used in chapter 3, to support different modes and ran it for each pair of modes. The resulting mean discovery times are shown in table 4.2. As expected, when both devices are in mode M_1, the smallest mean discovery time is obtained.

A second aspect of emulating the discovery process in the simulator is randomly drawing the discovery time between two devices at a particular encounter: while we know the *mean* discovery time as a function of the discovery modes used on the devices, in order to be accurate, we need to know the distribution of discovery times as well.

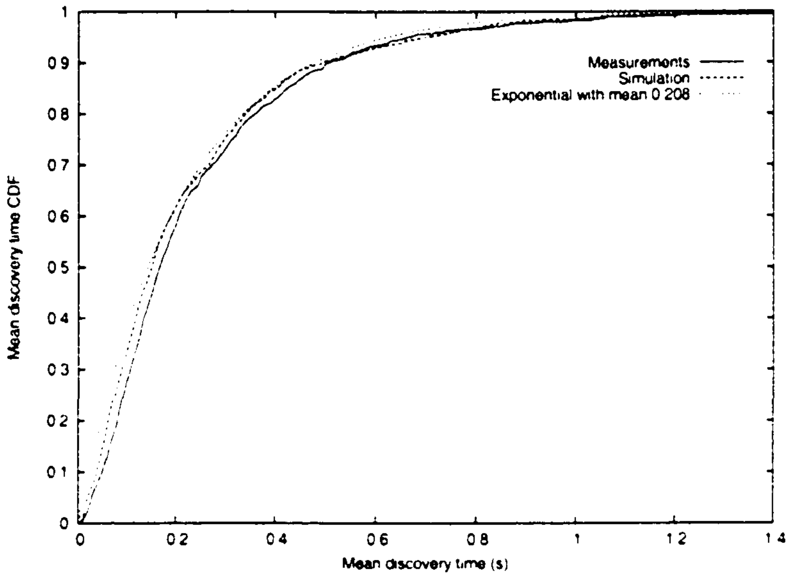


Figure 4.2: Discovery times times are exponentially distributed

Fortunately, it appears that at least when both devices are in mode M_1 the distribution of discovery times approximates very well the exponential distribution as can be seen in figure 4.2. We conjecture that this will be the case for other pairs of discovery modes and hence draw the discovery times from an exponential distribution with the respective mean (as per table 4.2).

We simplified our simulation by only considering pairwise contacts between devices: we do not implement the discovery process involving more than two devices that are in-range at the same time.

Mobility model

In our simulation, we tried to model the patterns of movement of people in the real world, that have certain preferred destinations or meeting areas. Hence, we employed the popular Random Waypoint model [5], augmented with *attraction areas*: square-shaped areas which nodes move towards with high probability. Nodes move in a rectangular 1000m x 1000m space. They choose a destination at random (but with high probability

Simulation field	1000m x 1000m
Attraction area size	20m x 20m
Speed	1m/s–5m/s (random uniformly distributed)
Pause time	2s–28s (random uniformly distributed)
Radio range	10m
Transfer speed	40 kB/s
Simulation length	100.000s (with 100.000s warmup)
Nodes	5–200 (depending on experiment)
Attraction areas	3-20 (depending on experiment)

Table 4.3: Simulation parameters

it will be in an attraction area) and move towards this destination at constant speed. Once the destination is reached, they pause for a random amount of time, after which they choose another destination and so on.

We run the simulation for 100.000 seconds as warmup (this is particularly important for the location-based scheme which has a learning phase) and then measure our statics over the next 100.000 seconds. The parameters of the simulation are shown in table 4.3.

4.3 Implementation

In this section, we describe the implementation of our two adaptive (or dynamic) schemes for selecting the discovery mode.

Recent activity level scheme

Figure 4.3 shows the implementation of the recent activity level scheme. By $t_{current}$ we denote the current time in the simulation, where as $t_{last\_contact}$ is the time of the last contact. $t_{nocontact}$ is a parameter of the algorithm: if $t_{nocontact}$ seconds have elapsed and

```
Made_Contact:
  t_lastcontact := t_current
End.


Update_Discovery_Mode:
    If (t_lastcontact = t_current)
        mode := 1
    Else If (t_current - t_lastcontact > t_nocontact)
        mode := mode - 1
End.
```

Figure 4.3: Recent-activity level scheme

no contact has been made the devices is switched down to the next lower (lazier) discovery mode. Whenever a contact is made the devices is switched up to the fastest discovery mode, M_1.

The default value of $t_{nocontact}$ is 5 seconds.

Location-based scheme

For the implementation of the location-based scheme, we use a grid to divide the simulation space into equal sized cells. Each device holds a counter, for each cell, of the number of contacts that were made while in that cell. The current discovery mode is chosen as a function of the contact counter of the current cell and the maximum contact counter of any cell as shown in figure 4.4. For example, suppose the device has previously discovered 2 devices in the current cell, while the highest number of contacts seen in any cell is 8. Then, the current discovery mode is switched to mode $M_{5-5*4/10} = M_3$.

The parameter of this scheme's implementation is the *cell size*, which determines the

```
Update_Discovery_Mode:
    mode := 5 - 5 * current_cell_counter / max_counter;
End.
```

Figure 4.4: Location-based scheme

granularity of the location-based decisions.

4.4 Results

In this section, we present the results of the experimental evaluation of the performance of the five schemes.

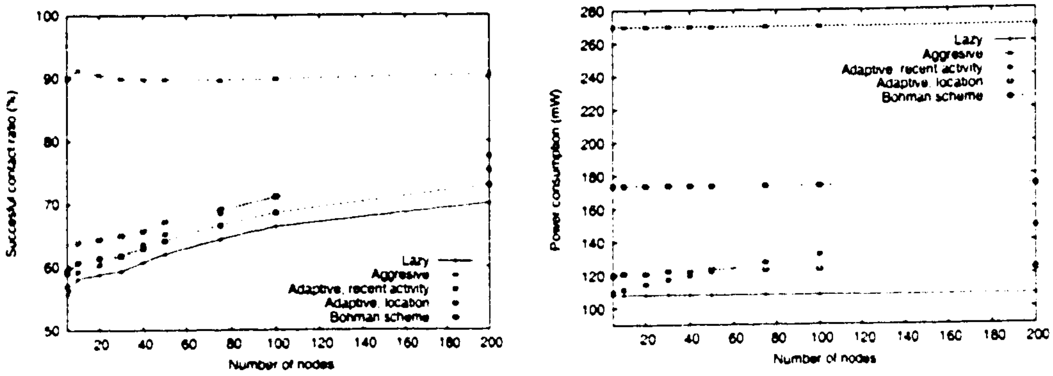
Communication opportunities

Each node in our simulation has a 10m radio range. Whenever two nodes get in-range, we count this occurrence as a communication opportunity and start the discovery process. If the nodes discover each other, we count a successful contact.

We measure the ratio of successful contacts for each of the five schemes while varying the number of nodes in the simulation, as shown in figure 4.5. Naturally, the static, aggressive scheme performs best in terms of contact ratio, but at a high cost in energy. The other schemes perform better as the density of nodes increases.

The "Bohman" scheme performs marginally better than the static, lazy scheme, but with much higher energy consumption. Our two adaptive schemes show promising results. The recent-activity scheme does best at higher node densities due to the increased frequency of contacts. On the other hand, the location-based scheme has almost constant power consumption regardless of node density.

Next, in figure 4.6, we plot the performance of the discovery schemes while varying the number of attraction areas. As the number of hubs increases the successful contacts



(a) Contacts ratio

(b) Power consumption

Figure 4.5: Performance of the five schemes versus number of nodes

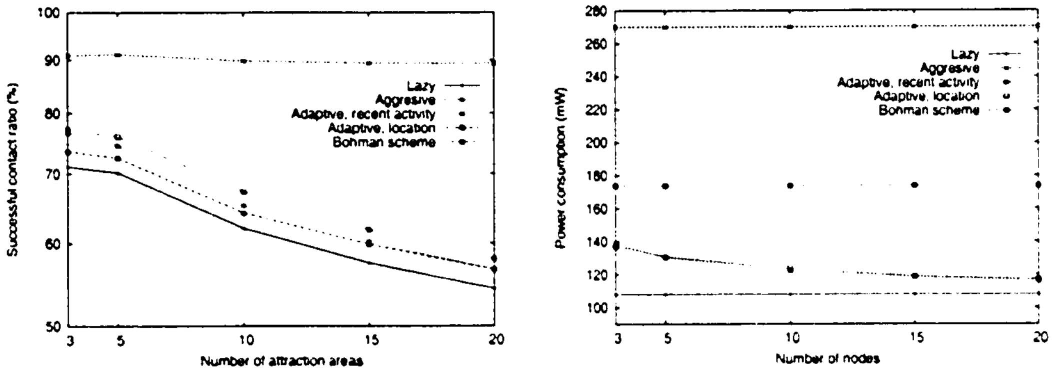
Scheme	Lazy	Aggressive	Bohman	Location	Activity
Energy/contact [mJ]	4.38	6.78	6.69	4.59	4.46

Table 4.4: Energy per successful contact

ratio decreases for all schemes, but the static-aggressive one. This can be accounted for due to the lower density of nodes when there are more attraction areas. A lower density of nodes implies that nodes meet more often while moving, rather than while they are pausing in an attraction area. Thus, windows of opportunity for contacts are shorter and more likely to be missed by the power-conserving scheme. The aggressive scheme, by virtue of having very short discovery times, is still able to discover other nodes, so its successful contacts ratio is almost constant.

In table 4.4, we derived another metric: the energy spent per successful contact for each of the five schemes. This metric was calculated for a particular parametrization of the simulation (10 attraction areas, 50 nodes). Our two adaptive schemes are almost as energy-efficient as the lazy scheme, and 50% more efficient than the aggresive scheme or the 'Bohman' scheme.

Next, we investigate the effect of varying the parameters of the adaptive schemes. Figure 4.7 shows the performance of the adaptive, recent-activity based scheme for different



(a) Contacts ratio

(b) Power consumption

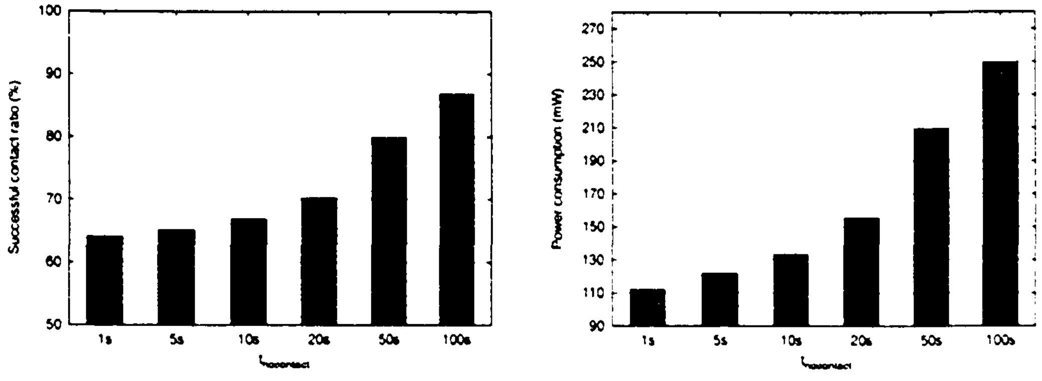
Figure 4.6: Performance of the five schemes versus number of attraction areas

values of the $t_{no\_contact}$ parameter. As expected, increasing the value of this parameter determines longer stays in aggressive states, hence better successful contact ratios, but higher power consumption as well. Clearly, a good choice for this parameter would depend on the parameters of the environment. In our case, a good value seems to be around 20 seconds, for which 70% of contacts are successful, for 150mW average power consumption (the idle mode itself consumes 90mW).

Figure 4.8 shows the performance of the location-based scheme while varying the cell size. Larger cell sizes mean that the effect of one encounter in the future spreads over a larger area. Interestingly, the effect of increasing the cell size seems to flatten out for values larger than 20m.

Data propagation

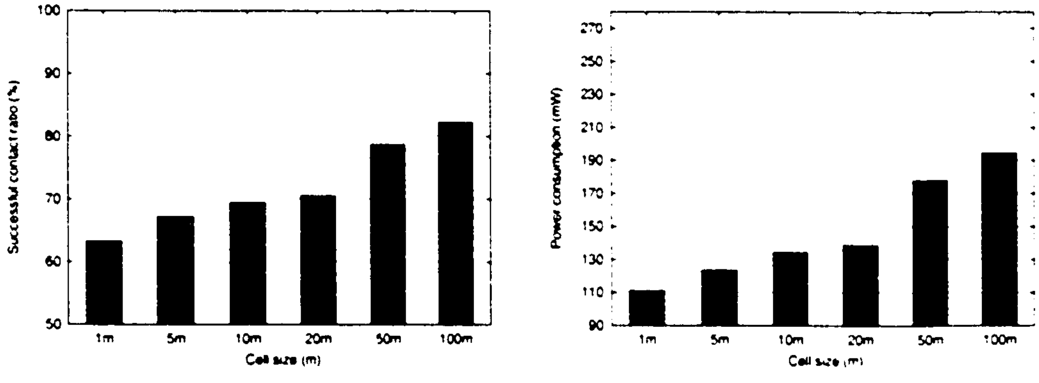
There is a double benefit of being engaged in a faster discovery mode when there is another device in range. In addition to increasing the chances of a successful contact, discovering a neighbouring device in a shorter time allows the devices more time for data exchange. In this section, we investigate the effect of the discovery mode selection on the throughput of our ad hoc network.



(a) Contacts ratio

(b) Power consumption

Figure 4.7: Performance of the adaptive, recent-activity scheme versus $t_{nocontact}$



(a) Contacts ratio

(b) Power consumption

Figure 4.8: Performance of the adaptive, location-based scheme versus *cell size*

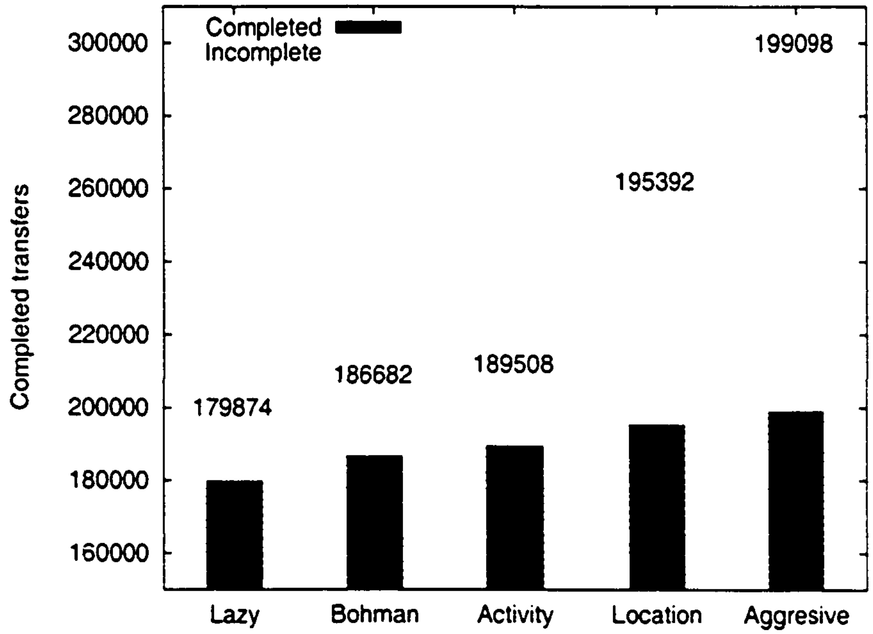


Figure 4.9: Throughput of the five schemes

We modify the simulation, so that at each encounter between two nodes, after they discover each other, they transfer a file of 100kB in size at the typical Bluetooth transfer rate of 40kB/s (yielding a transfer time of 2.5 seconds). A transfer is successful finishes if the two nodes do not walk out of range sooner than 2.5 seconds after discovery. We run the simulation with 50 nodes and 10 attraction areas and count the number of completed and incomplete transfers. Results are shown in figure 4.9.

Both adaptive schemes show better performance than the Bohman and lazy schemes. The location-based scheme outperforms the Bohman and lazy schemes, by 4.6% and 8.6% respectively. Its performance is very close to the fastest scheme, while its power consumption, as we have seen in section 4.4 is close to the energy-efficient scheme.

Chapter 5

Related Work

Opportunistic Networking Previous work has looked into the concept of networking in the absence of infrastructure by using node mobility for data propagation. Several terms have been proposed to describe this type of communication: opportunistic networking [28], Pocket Switched Networks (PSN) [7, 13], Intermittently Connected Mobile Ad Hoc Networks (ICMAN) [12]. Opportunistic networking is considered to fall under the larger category of Delay Tolerant Networking (DTN) [6],i.e., networking in the absence of end-to-end connectivity.

Epidemic Propagation Much of the work in DTNs has been focused on the opportunistic forwarding of data in the absence of end-to-end paths. This process has been referred to as data muling [24] or store-and-haul forwarding [27]. Several schemes have been proposed for data forwarding. Vahdat *et al.* [29] propose epidemic propagation: a flooding-style algorithm where nodes forward messages to every node they meet. In Message Ferrying [32], highly mobile nodes called ferries take on the task of carrying messages between disconnected mobile nodes. Examples of real environment networks that use opportunistic forwarding to relay data collected by sensors carried by animals include ZebraNet [15, 18] and SWIM [26]. These papers focus on the routing and forwarding mechanisms and do not address the question of neighbour discovery explicitly; neighbor discovery is a key step in establishing opportunistic communication.

User studies of mobility patterns Murphy *et al.* [19] use measurements of the discovery time and the range of Bluetooth devices to investigate the feasibility of using Bluetooth for inter-vehicular networking. This study does not consider symmetric neighbour discovery. Rather, the measurements of discovery time were done in an asymmetric fashion with one device in inquiry mode and the other in scan mode. Su *et al.* [28] gathered traces of human mobility patterns during two user studies involving users carrying PDAs with Bluetooth. One limitation of their study is that brief contacts, such as those that occur when two users walk past each other, may be missed due to the way neighbour discovery is performed. The same limitation applies to two other studies [7, 13] which gather data on contact frequency and duration between users carrying iMotes equipped with Bluetooth. The granularity at which the neighbour discovery process is run, i.e., every 2 minutes, implies that most brief contacts are not logged.

Bluetooth (asymmetric) discovering procedure The Bluetooth discovering (inquiry) procedure has been extensively studied [30, 31, 16, 20, 9]. Kasten *et al.* [16] perform measurements of the discovery time for Bluetooth v1.1 devices. Other authors [31, 20] derive analytical formulas for the mean discovery time between two devices and validate them through simulation. All of these papers consider only the standard asymmetric scenario where one of the devices is listening and the other is sending beacons.

Symmetric neighbour discovery Alonso *et al.* [1] study the generic neighbour discovery problem in ad-hoc networks analytically. For Bluetooth, several authors [21, 8, 25] propose and analyze symmetric neighbour discovery schemes that involve modifying the standard [3] Bluetooth protocol. By contrast, our scheme works on currently available hardware.

Two studies [22, 4] have considered neighbour discovery in the unmodified Bluetooth protocol. Salonidis *et al.* [22] have considered a symmetric discovery scheme for Bluetooth v1.1. They were the first to point out the need for alternating between the inquiry and scan modes in corresponding inquire/scan phases in a peer-to-peer setting. They have also shown that the time spent performing each phase should be random in order to ensure bounded discovery times. Their paper is based on an analytical evaluation of the Bluetooth discovery protocol with some inherent simplifications, such as, not taking into account the two different inquiry trains in the discovery process. According to further studies [4, 16], the estimated discovery time could not be achieved experimentally. Salonidis *et al.* [22] also considered that a device is scanning continuously during the scan phase and not only for the duration of a window out of an scan phase interval. This implies a 100% slot usage for the discovery process.

Bohman *et al.* [4] have further studied the Bluetooth discovery scheme by means of simulation and measurements on an experimental testbed. They have defined the *residence time*, i.e. the time a device performs a given role as composed of a fixed and a variable part distributed either uniformly or exponentially. The authors conclude that there is little difference between using a uniform or exponential distribution based on simulation and experimental results. They also estimate the time necessary for establishing communication using symmetric discovery to be at least 8 seconds when the mean residence time in each role is around 2.5 seconds.

No previous work reports on synchronization artifacts that appear when devices (or their simulation abstractions) are not correctly initialized by randomizing the corresponding instants of their activation.

Mobility simulations The Random Waypoint model has been widely used for ad-hoc network simulations [5] including in opportunistic forwarding simulations [29]. Ghosh *et al.* [12] base their epidemic forwarding algorithms on the idea of *sociological orbits*: the repetitive patterns of movement typical of everyday life.

Energy efficiency Galluccio *et al.* [11] has proposed an analytical framework for studying the tradeoff between energy efficiency and time for the neighbour discovery process in ad-hoc networks. In [23], Sedov *et al.* discuss an approach for making the service discovery protocol of Bluetooth more energy-efficient.

Chapter 6

Conclusions

In this thesis, we present the first study of Bluetooth neighbour discovery, not only from the point of view of short discovery times, but also from the perspective of power consumption. We introduce a novel scheme for symmetric neighbour discovery in Bluetooth which extends previous studies by exploring using different mean residence times and different timing for the scan windows. We use a discrete-event simulator to explore the parameter space and we validate our results by experimental measurements on real devices to evaluate the performance of this scheme, in terms of speed and energy-efficiency. Our study rigorously sets up simulations and measurements so that synchronization artifacts are not introduced and the randomness of the system is guaranteed. No previous work that we are aware of has considered randomizing experiments as an important aspect of validation. Our work introduces a systematic methodology for experiment randomization to set the instant at which two devices come into range in a statistically meaningful way.

Our results show a two-fold improvement in both mean discovery time and power consumption over the previous work. Our fastest discovery mode has a mean discovery time of 0.2 seconds. For a reasonable discovery time of 1 second, the power consumption of the device is as low as 1.5 times the power consumption of the idle mode.

Our evaluation of the neighbour discovery scheme forms the basis on top of which further discovery algorithm extensions or intelligent policies for neighbor communication establishment can be built. For example, in the target scenario of ad-hoc opportunistic Bluetooth networks an obvious optimization is to adapt the power usage of the neighbour discovery process according to the probability of discovery success.

To this end, we propose two algorithms that adaptively switch between discovery modes that are economic and those that have fast discovery times in order to maximize battery life and the chances of discovering neighbouring devices at the same time. One algorithm uses the recent level of activity and the other uses the location of previous encounters to predict the probability of having a nearby device.

We evaluate these algorithms through a node mobility simulation. Our two adaptive schemes spend 50% less energy per contact and have 4.6% and 8.6% better performance, respectively over a naïve, power-conserving scheme.

While the simulation results are promising, there are inherent limitations in modeling real-world patterns of movement through a simplified mobility model. In the future, we intend to extend our work by evaluating our adaptive schemes through user studies. While there have been similar studies of user mobility in the context of opportunistic networking, they were limited in tracking brief contacts, due to the low frequency of the discovery process. Our work provides fast and energy-efficient discovery modes, so that even short contacts can be logged, without excessive power consumption.

In conclusion, this thesis presents the first proof of feasibility for the use of Bluetooth for opportunistic networking. In addition, our algorithms and techniques work on the unmodified Bluetooth standard so they can be readily implemented on currently deployed devices.

Bibliography

[1] G. Alonso, E. Kranakis, C. Sawchuk, R. Wattenhofer, and P. Widmayer. Randomized Protocols for Node Discovery in Ad-hoc Multichannel Broadcast Networks. In *Proceedings of 2nd Annual Conference on Adhoc Networks and Wireless (ADHOC-NOW'03)*, pages 104–115, Montreal, Canada, Oct 2003.

[2] Bluetooth SIG. Bluetooth Baseband Specification - Version 1.1, 2000.

[3] Bluetooth SIG. Specification of the Bluetooth System - Version 1.2. Specification Volumes 0-3, 2003.

[4] D. Bohman, M. Frank, P. Martini, and Scholz C. Performance of Symmetric Neighbor Discovery in Bluetooth Ad Hoc Networks. In *GI Jahrestagung*, volume 1, pages 138–142, 2004.

[5] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom '98)*, Dallas, TX, Oct 1998.

[6] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay tolerant network architecture, draft-irtf-dtnrg-arch-02.txt, July 2004.

[7] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket Switched Networks: Real-World mobility and its Consequences for Opportunistic Forwarding. Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Lab, 2005.

[8] K. Cheolgi, M. Joongsoo, and L. Joonwon. A Random Inquiry Procedure using Bluetooth. In *Proceedings of the International Conference on Communication in Computing (CIC)*, Las Vegas, NV, Jun 2001.

[9] M. Duflot, M. Kwiatkowska, G. Norman, and D. Parker. A Formal Analysis of Bluetooth Device Discovery. In *Proceedings of the 1st International Symposium on Leveraging Applications of Formal Methods (ISOLA'04)*, 2004.

[10] E. Ferro and F. Potorti. Bluetooth and Wi-Fi Wireless Protocols: A Survey and A Comparison. *IEEE Wireless Communications*, 12(1):12–26, Feb 2005.

[11] L. Galluccio, G. Morabito, and S. Palazzo. Analytical evaluation of a tradeoff between energy efficiency and responsiveness of neighbor discovery in self-organizing ad hoc networks. *IEEE Journal on Selected Areas in Communication*, 22(7):1167–1182.

[12] J. Ghosh, S. Philip, and C. Qiao. Sociological Orbit Aware Location Approximation and Routing in MANET. In *Proceedings of the IEEE International Conference on Broadband Networks (BROADNETS)*, Boston, MA, Oct 2005.

[13] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket Switched Networks and Human Mobility in Conference Environments. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN '05)*, pages 244–251, Philadelphia, PA, Aug 2005.

[14] S. Jain, K. Fall, and R. Patra. Routing in A Delay Tolerant Network. In *Proceedings of ACM SIGCOMM*, 2004.

[15] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, pages 96–107, San Jose, CA, Oct 2002.

[16] O. Kasten and M. Langheinrich. First Experiences with Bluetooth in the Smart-Its Distributed Sensor Network. In *Workshop on Ubiquitous Computing and Communications (PACT 2001)*, Oct 2001.

[17] J. LeBrun, C.-N. Chuah, and D. Ghosal. Knowledge Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks. In *Proceedings of IEEE Vehicular Technology Conference (VTC)*, pages 1–5, 2005.

[18] T. Liu, C.M. Sadler, P. Zhang, and M. Martonosi. Implementing Software on Resource-Constrained Mobile Sensors: Experiences with Impala and Zebranet. In *Proceedings of the Second International Conference on Mobile Systems, Applications and Services (MobiSys 2004)*, pages 256–269, Boston, MA, Jun 2004.

[19] P. Murphy, E. Welsh, and J. Frantz. Using Bluetooth for Short-Term Ad Hoc Connections Between Moving Vehicles: A Feasibility Study. In *Proceedings of the 55th IEEE Vehicular Technology Conference*, volume 1, pages 414–418, Birmingham, AL, May 2002.

[20] B. Peterson, R. Baldwin, and J. Kharoufeh. A Specification-Compatible Bluetooth Inquiry Simplification. In *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37)*, 2004.

[21] M.A. Rónai and E. Kail. A Simple Neighbour Discovery Procedure for Bluetooth Ad Hoc Networks. In *Proceedings of IEEE Globecom 2003*, San Francisco, CA, Dec 2003.

[22] T. Salonidis, P. Bhagwat, and L. Tassiulas. Proximity Awareness and Fast Connection Establishment in Bluetooth. In *Proceedings of the First Annual ACM Workshop on Mobile and Ad Hoc Networking and Computing*, Boston, MA, August 2000.

[23] I. Sedov, M. Haase, S. Preuss, C. Cap, and D. Timmermann. Time and Energy Efficient Service Discovery in Bluetooth. In *Proceedings of the 57th IEEE Vehicular Technology Conference (VTC)*, Jeju, Korea, 2003.

[24] R.C. Shah, S. Roy, S. Jain, and W. Brunette. Datamules: Modelling a Three Tiered Architecture for Sparse Sensor Networks. In *Proceedings of First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, May 2003.

[25] F. Siegemund and M. Rohs. Rendezvous Layer Protocols for Bluetooth-Enabled Smart Devices. In *Proceedings of the International Conference on Architecture of Computing Systems (ARCS02)*, pages 256–273, Karlsruhe, Germany, 2002.

[26] T. Small and Z.J. Haas. The Shared Wireless Infostation Model: A New Ad Hoc Networking Paradigm (or Where There Is A Whale, There Is A Way), booktitle = "proceedings of the 4th acm international symposium on mobile ad hoc networking & computing (mobihoc '03)", year = "2003", address = Annapolis, MD, pages="233–244".

[27] J.P. Sterbenz, R. Krishnan, R.R. Hain, A.W. Jackson, D. Levin, R. Ramanathan, and J. Zao. Survivable Mobile Wireless Networks: Issues, Challenges, and Research Directions. In *Proceedings of the 3rd ACM workshop on Wireless security (WiSE '02)*, Atlanta, GA, 2002.

[28] Jing Su, Alvin Chin, Anna Popivanova, Ashvin Goel, and Eyal de Lara. User Mobility for Opportunistic Ad-Hoc Networking. In *6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 2004.

[29] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. Technical Report CS-200006, Duke University, Apr 2000.

[30] G. Zaruba. Accelerated Neighbor Discovery in Bluetooth Based Personal Area Networks. In *Proceedings of the 2002 International Conference on Parallel and Distributed Processing Techniques and Applications*, 2002.

[31] G. Zaruba and V. Gupta. Simplified Bluetooth Device Discovery - Analysis and Simulation. In *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37)*, 2004.

[32] W. Zhao, M. Ammar, and E. Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In *Proceedings of the 5th ACM International Symposium on Mobile Ad-Hoc Networking and Computing*. ACM Press, 2004.