Adaptive Energy Conserving Algorithms for Neighbor Discovery in Opportunistic Bluetooth Networks

Catalin Drula, Cristiana Amza, Franck Rousseau, and Andrzej Duda

***Abstract*— In this paper, we introduce and evaluate novel adaptive schemes for neighbor discovery in Bluetooth-enabled ad-hoc networks. In an ad-hoc peer-to-peer setting, neighbor search is a continuous, hence battery draining process. In order to save energy when the device is unlikely to encounter a neighbor, we adaptively choose parameter settings depending on a mobility context to decrease the expected power consumption of Bluetooth-enabled devices. For this purpose, we first determine the mean discovery time and power consumption values for different Bluetooth parameter settings through a comprehensive exploration of the parameter space by means of simulation validated by experiments on real devices. The fastest average discovery time obtained is 0.2 s, while at an average discovery time of 1 s the power consumption is just 1.5 times that of the idle mode on our devices. We then introduce two adaptive algorithms for dynamically adjusting the Bluetooth parameters based on past perceived activity in the ad-hoc network. Both adaptive schemes for selecting the discovery mode are based only on locally-available information. 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 static power-conserving scheme.**

I. Introduction

MOBILE devices with short range wireless network interfaces become increasingly popular and enable new peer-to-peer applications involving devices communicating in proximity of each other. One example of such applications are ad hoc *opportunistic networks* [1], [2], [3], [4] in which content is forwarded between mobile devices in the absence of global connectivity 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.

Manuscript received December 15, 2005; revised July 15, 2006. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), IBM Centers of Advanced Study (IBM CAS) and IBM Research, Communications and Information Technology Ontario (CITO), Canadian Foundation for Innovation (CFI) and Institut National Polytechnique de Grenoble (INPG)

Catalin Drula is with the Department of Computer Science, University of Toronto, Canada. Email: catalin.drula@utoronto.ca.

Cristiana Amza is with the Department of Electrical and Computer Engineering, University of Toronto, Canada. Email: amza@eecg.toronto.edu.

Franck Rousseau and Andrzej Duda are with the INPG-ENSIMAG, LSR-IMAG Laboratory, Grenoble, France. Email: {rousseau, duda}@imag.fr

Bluetooth-enabled mobile devices are particularly suitable for this kind of new communication peer-to-peer applications, because they offer low-power short-range data transfers [5]. 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, neighboring devices first need to discover each other, i.e., learn about each other's presence. This implies that a continuous, or at least periodic, process for scanning the surroundings i.e., the *neighbor discovery process*, must be run on each device. Due to its continuous nature, the discovery process may consume considerable amounts of energy, even if individual data transfers on Bluetooth are low-power. Previous user studies [1] have shown that the energy consumption of the Bluetooth discovery protocol is the limiting factor for wide deployment of such ad-hoc Bluetooth networks, because mobile devices have limited battery life.

In this paper, we first explore the inherent trade-off in the selection of Bluetooth parameters between energy efficiency and neighbor discovery speed. Specifically, depending on the choice of Bluetooth 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.

Based on our exploration of Bluetooth parameter settings, we introduce and evaluate two novel adaptive algorithms for dynamically selecting the parameters of the neighbor discovery process in order to switch between low-power, slow discovery modes and high-power, fast discovery modes, depending on a mobility context. Both adaptive schemes for selecting the discovery mode are based only on locally available information. In particular, we use the real-life observation that wherever a neighbor device is encountered, other contacts are likely to happen as well. Such is the case, for example in classrooms, shopping complexes, public transport, etc. Our first scheme uses 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.

In our evaluation, we first explore the energy consumption versus neighbor discovery speed trade-off in Bluetooth and we select a set of representative parameter settings. We then evaluate our two adaptive discovery selection schemes, using dynamic parameter settings in a mobility simulation. In order to efficiently search the multi-dimensional Bluetooth 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 confirm the discovery time and power consumption for important parameter settings (e.g., yielding minimum discovery times or a good compromise between discovery time and power consumption) 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. Furthermore, 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 fastest average discovery time obtained from our systematic Bluetooth parameter setting search is 0.2 seconds. While several previous studies have considered neighbor discovery in Bluetooth version 1.1 [6], [7], [8], the best previous *experimentally validated* discovery scheme reports a minimum discovery time of 8 seconds [7]. Another contribution of our work is to find parameter settings that dramatically improve *both* Bluetooth discovery time and power consumption at the same time. For an average discovery interval of 1 second, which is well within the usual window of opportunity for establishing contact, the power consumption can be as low as 1.5 times the idle mode power consumption. To our knowledge, this is the first result to show feasibility of a fast neighbor discovery scheme on currently available hardware in the envisioned ad-hoc Bluetooth network setting.

To evaluate our adaptive neighbor discovery selection schemes, we use a mobility simulator implementing a modified Random Waypoint [9] model to evaluate the performance of our two adaptive schemes along with the performance of static discovery mode schemes. Our simulation results show good performance for the two adaptive schemes when compared to the static 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 our 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.

The rest of this paper is organized as follows. Section II provides the background on the discovery process in Bluetooth. In Section IV, we present our two schemes for dynamically selecting the discovery mode in Bluetooth in an opportunistic ad-hoc network. In Section III, we explain how we perform the search for the optimal parameters through simulation, how we validate the simulation through measurements on real devices, and we present other aspects of our simulation methodology, such as the simulation of mobility in ad-hoc opportunistic Bluetooth networks. Section V presents our evaluation of the adaptive and static schemes. We discuss the related work in Section VI and conclude in Section VII.

II. BACKGROUND

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

A. 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 [2], [1] investigate the idea of *opportunistic networking* or, alternatively called, *pocket switched networking* [3]: 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 in which this type of networking might be useful are remote and rural areas with sporadic access to broadband infrastructure [10]. Another potential use would be networking when access infrastructure has failed, such as under disaster conditions. Another study [1] 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 *neighbor 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:

1) *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 10 m such as that of Bluetooth and walking towards each other at a normal speed of 2 m/s will have a window of opportunity of only 5 s 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.

2) *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.

B. Standard Asymmetric Bluetooth Discovery Procedure

Typical Bluetooth [11] 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 that consists in listening to beacons and sending responses.

Figure 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 1). The inquiring device uses the frequencies only in one of the trains at a time. It switches trains every 2.56 s. 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 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:

$$
\begin{aligned}
F_{inquiry} &= [CLK_{16-12} + k_{\text{offset}} + (CLK_{4-2,0} \\
&\quad - CLK_{16-12}) \bmod 16] \bmod 32 \\
F_{scan} &= [CLK_{16-12} + N] \bmod 32 \qquad (1)
\end{aligned}
$$

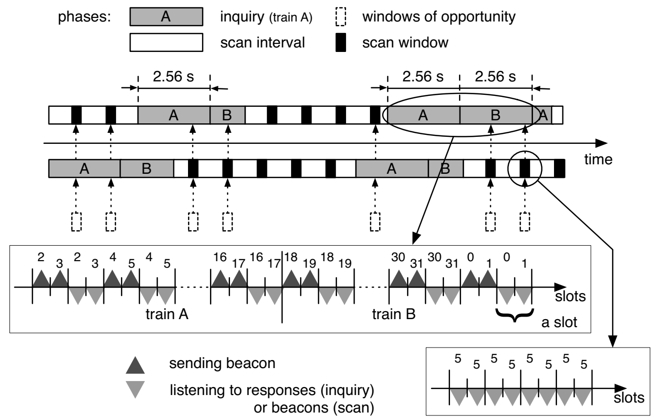


Fig. 1. Bluetooth discovery process

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 s, 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 the 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 a slight benefit of avoiding collisions, that are anyway fairly unlikely [8]. 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.

C. Symmetric Neighbor Discovery in Bluetooth-enabled Ad-Hoc Networks

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 [12] that have considered neighbor discovery in Bluetooth v1.1. Salonidis *et al.* [12] have shown that the *residence time* during which a device performs a given role (active or passive), should be random in order to ensure bounded discovery times. Otherwise, if the residence times are deterministic, two devices with synchronized schedules for alternating roles would never discover each other.

Bohman *et al.* [7] have further studied the Bluetooth symmetric 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, randomly chosen part distributed either uniformly or exponentially. They have assumed equal residence times and default scan window timing and by trying different parameters for the residence time, they found that when the mean residence time in each role is around 2.5 s, 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 s. They also concluded that there is almost no difference between using a uniform or exponential distribution for the variable part of the residence time.

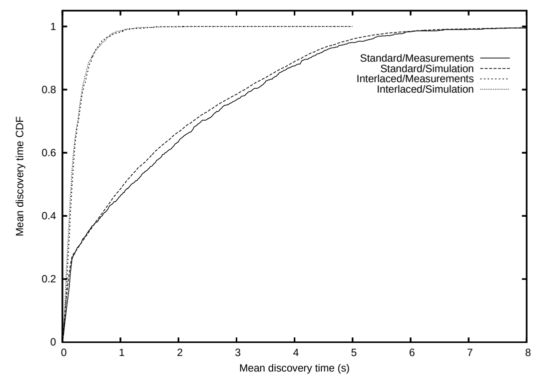


Fig. 2. Cumulative distribution functions of the discovery time

III. Basic neighbor discovery scheme

In this section, we presents the basic neighbor discovery scheme and show how to set its parameters by exploring a large space of possible values. We thus can understand the trade-off between power consumption and discovery time on our devices that will be further used in the selection of our adaptive discovery modes.

We extend the approach proposed by Bohman *et al.* [7]—we assume that the residence time spent by a device in each of the two phases (inquiry and inquiry scan) during the discovery process are as follows:

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

where C_{inq}, V_{inq}, C_{scan}, V_{scan} are respectively, parameters of the constant and the variable part of the residence time in each phase, $\mathrm{rand}(x, y)$ denotes an integer-valued uniformly distributed random variable with values in the interval (x, y) (note that the mean values of the variable parts are thus V_{inq} and V_{scan}). We also propose to vary $scan\_window$ and $scan\_interval$, the duration and frequency of the scan windows (how often and for how long the passive device is listening while in the scan phase). We consider the Bluetooth v1.2 protocol and explore the use of the interlaced inquiry scan mode in addition to the standard one. Our scheme does not require any modification of the standard and can be implemented on the currently available hardware.

The basic scheme comes from considering two main performance objectives: fast discovery and low power consumption. We look for two-fold benefits: a reduction in power consumption and an improvement of the mean discovery time. 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 during which the device is only active for a fraction of time (the scan window). Hence, the key to finding the best of the two worlds (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 scan window. Choosing the right values of the mean residence times as well as the frequency and duration of the scanning window thus enables us to find parameter settings offering an energy efficient scheme with a short mean discovery time.

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 depends on the type of activity performed in a given time slot.

A. Parameter Space Search

Since the parameter space is seven-dimensional, it would be infeasible 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 [11] with the specific features of the Bluetooth devices that we used in our measurement. 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. For the experiments, we use two Cellink BTA-3100 devices with a Silicon Wave chipset and compatible with v1.2 of the Bluetooth protocol.

Table I 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 size 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 I).

Figure 2 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 through simulation. We also observe that the interlaced mode performs noticeably better due to its scanning of frequencies in both inquiry trains.

Furthermore, in our parameter space search, we are interested in capturing the power consumption of the discovery process versus discovery time trade-off. We derive 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 I

PARAMETER SPACE EXPLORED IN SIMULATION

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

TABLE II

POWER CONSUMPTION OF THE CELLINK BTA-3100 DEVICE

consumption from the parameters of the discovery process based on careful measurement of our operational prototype on the real devices. 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, using fewer slots for discovery implies having 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 mean only power spent in excess of the idle mode consumption.

We have measured the power consumption of our Cellink devices in the idle mode and in the two discovery states (scan and inquiry) as shown in Table II. This was done by placing a precision resistor in series with the Bluetooth device and measuring the voltage drop across this resistor with the aid of a digital storage oscilloscope. As we can see from this Table, the inquiry scan mode uses considerably more power than the inquiry mode on the devices in our experimentation platform.

In order to visualize all relevant points in the power consumption versus discovery time trade-off, we plot the graph shown in Figure 3 as follows. For each value of the mean discovery time obtained in the simulations, we retain the point that has the minimum power usage of the discovery process. We then use these values to draw the power vs. discovery time graph with the power expressed in milliwatts as shown in Figure 3. We observe from this 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. As expected, spending more power results in shorter discovery times. Interlaced scanning outperforms standard scanning over the whole range of parameters of interest.

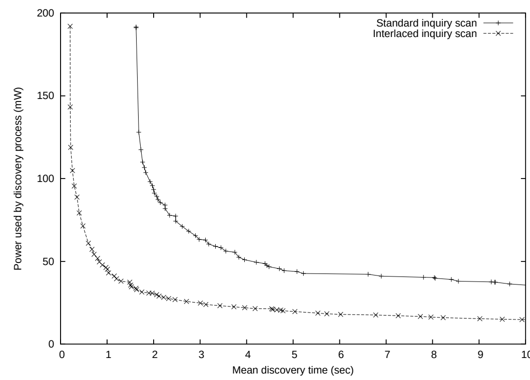


Fig. 3. Mean discovery time vs. power consumption of the discovery process

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 III

PARAMETER VALUES FOR GOOD TRADE-OFF POINTS (SIMULATION)

Table III 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).

Discovery Modes Used in the Opportunistic Ad-Hoc Network Simulation

We chose five discovery modes for use in our ad-hoc network simulation as shown in Figure 4: M_1 is the most aggressive mode and M_5 is the laziest one. At any given time, a mobile device in our simulation uses one of these discovery modes. Table IV 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 state-of-the-art scheme of Bohman *et al* [7],

	$scan\_interval$[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 IV

PARAMETERS FOR DISCOVERY MODES USED IN THE SIMULATION

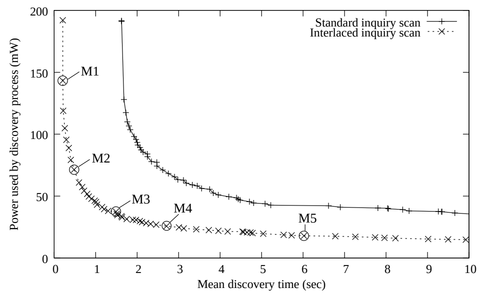


Fig. 4. Discovery time versus power consumption curve. M1 through M5 are discovery modes we use in the ad-hoc network simulation.

	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 V

MEAN DISCOVERY TIMES BETWEEN TWO DEVICES IN DIFFERENT MODES [s]

which we call M_b.

In the previous subsection, we have 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 have to consider the case in which two nodes in range use different discovery modes. Hence, we modified our discrete-event simulator to support different modes and we ran it for each pair of modes. The resulting mean discovery times are shown in Table V. 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. Fortunately, it appears that at least when both devices are in mode M_1, the exponential distribution approximates very well the distribution of discovery times as it can be seen from Figure 5. Hence, we draw the discovery times from an exponential distribution with the respective mean (as per Table V).

Finally, we simplify 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.

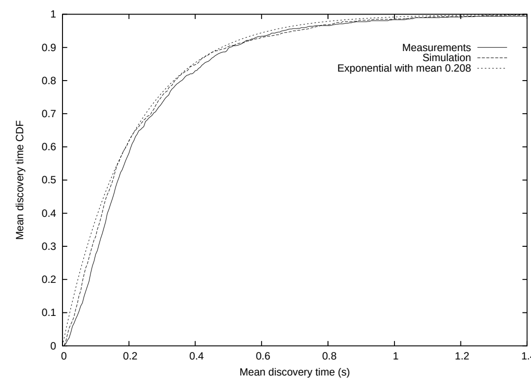


Fig. 5. Discovery times are exponentially distributed

IV. ADAPTIVE DISCOVERY MODE SELECTION

We describe in this section two adaptive schemes for switching between different discovery modes.

A. Overview

From the point of view of the trade-off that discovery modes achieve between discovery time and power consumption, we could classify them as ranging from *aggressive* (fast discovery, energy intensive) to *lazy* (slow discovery, low power consumption). It is clear that in general no single discovery mode represents the best choice. Aggressive modes would drain the battery of the mobile device too soon, while lazy modes would miss many communication opportunities. Thus, the choice of a discovery mode should be user-specified according to the priorities at a given instant: 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 introduce two schemes for dynamically switching discovery modes based solely on local knowledge. The two schemes are based on the observation previously made by Ghosh *et al.* [13] 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 the variant of this observation that "wherever a device is encountered, it is likely that there are more devices".

```
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.
```

Fig. 6. Recent-activity level scheme

Next, we describe our two adaptive schemes based on the recent activity level and recognizing locations of past contacts, respectively:

a) 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.

b) Location of 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.

V. Evaluation of the Adaptive Schemes

In this section, we evaluate by means of simulation the proposed adaptive schemes for selecting the discovery modes. We begin with a presentation of their algorithms as implemented in the simulation and we describe the simulation environment. Finally, we discuss the simulation results.

Recent activity level scheme

Figure 6 shows the algorithm of the recent activity level scheme. By $t_{current}$ we denote the current time in the simulation, $t_{last\_contact}$ is the time of the last contact, and $t_{nocontact}$ is a parameter of the algorithm: if $t_{nocontact}$ seconds have elapsed and no contact has been made the devices switch to the next lazier discovery mode. Whenever a contact is made, the devices switch to the fastest discovery mode—M_1. The default value of $t_{nocontact}$ is 5 s.

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 7. For example, assume that 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 $M_{5-5*4/10} = M_3$. *cell size* is a parameter of this scheme that determines the granularity of the location-based decisions.

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

Fig. 7. Location-based scheme

Other schemes

We also introduce three simple static schemes for comparison with our dynamic schemes:

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 proposed by Bohman et al. [7] (i.e. discovery that uses equal residence times and default scan window timing).

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

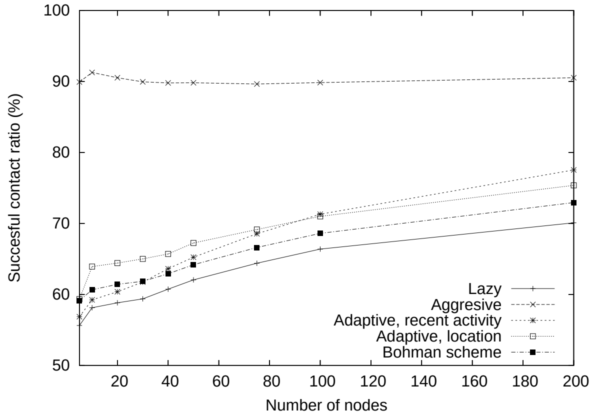
c) Successful contacts ratio: Ideally, all possible communication opportunities, i.e., when two device are in range, then contact is actually established. In practice, however, if the discovery time is greater than the time when the devices are in range, they will not discover each other. Such an occurrence 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.

d) 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 neighboring 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.

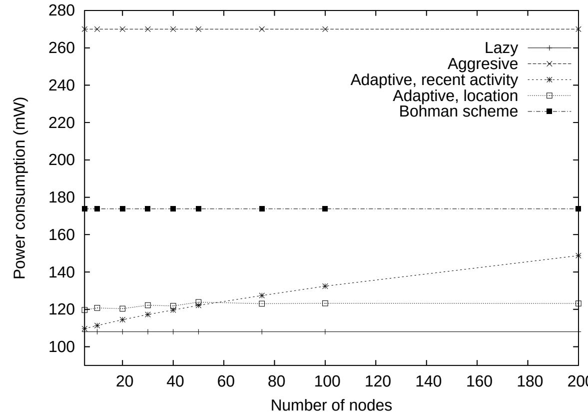
Both metrics of performance must be weighed against the energy consumption (otherwise, clearly the most aggressive discovery mode outweighs all the others in performance). Hence, in our simulation, we measure average power consumption of the devices and present it alongside the performance results.

Simulation environment

In our simulation, we model the patterns of real world movement of people that have certain preferred destinations or meeting areas. We have used the familiar Random Waypoint



(a) Contact ratio



(b) Power consumption

Fig. 8. Performance of the five schemes versus number of nodes

model [9] augmented with *attraction areas*—square-shaped areas towards which nodes move with high probability. Although this model presents several known shortcomings (it may generate unrealistic movement patterns such as “sharp turns” or “sudden stops”, it requires long initial warm up sequences, it slowly converges to the stationary distribution [14]), it is simple and allows us to get an insight into the performance behavior of the proposed schemes. Obviously, an ideal simulation would use real-world traces, but this requires access to representative data sets and a complex simulation set up. Compared with mobility models derived from traces, the Random Waypoint model may introduce a significant bias; for example in a recent comparison with a model based on WLAN traces, the following results have been reported: the average relative speed was one order of magnitude higher, the average link duration was approximately double, and the average spatial density was comparable to that of the Random Waypoint model [15]. However, comparisons of different mobility models also show that the Random Waypoint yields comparable performance metrics concerning the network topology and ad hoc routing performance as the recent Obstacle Mobility model [16].

In our simulation, nodes move in a rectangular 1000 m x 1000 m space. They choose a destination at random (but with high probability 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 s as a warmup (this is particularly important for the location-based scheme that has a learning phase) and then gather our statistics over the next 100,000 s. The parameters of the simulation are shown in Table VI.

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

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 VI
SIMULATION PARAMETERS

Communication opportunities

Each node in our simulation has a 10 m 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 8. Naturally, the static aggressive scheme performs best in terms of contact ratio, but at the expense of a higher energy consumption. 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. We can see that the proposed schemes perform well even for sparse node density: for a large range of the number of nodes in the same region, the percentage of successful contacts stays relatively flat—it varies from 60 % to 75 %. The recent-activity scheme does the 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 the node density.

Next, in Figure 9, we plot the performance of the discovery schemes while varying the number of attraction areas. As the number of hubs increases, the successful contact ratio decreases for all schemes, except for the static aggressive one. This can be accounted for 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

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

TABLE VII

ENERGY PER SUCCESSFUL CONTACT

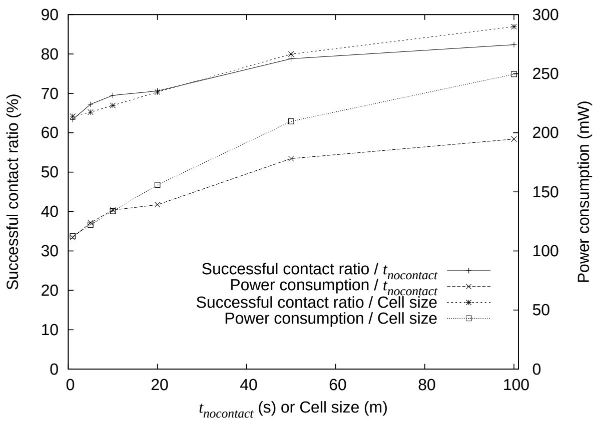


Fig. 10. Performance of the adaptive recent-activity scheme versus $t_{nocontact}$ and performance of the adaptive location-based scheme versus *cell size*

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 contact ratio is almost constant.

In Table VII, we show 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 aggressive scheme or the 'Bohman' scheme.

Next, we investigate the effect of varying the parameters of the adaptive schemes. Figure 10 shows the performance of the adaptive recent-activity based scheme for different values of $t_{nocontact}$ 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 s for which 70% of contacts are successful along with 150 mW average power consumption (the idle mode itself consumes 90 mW).

Figure 10 also shows the performance of the location-based scheme while varying the cell size. Larger cell sizes mean that the effect of one encounter spreads over a larger area in the future. Interestingly, the effect of increasing the cell size seems to flatten out for values larger than 20 m.

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 neighbor device in a shorter time allows the devices to have 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.

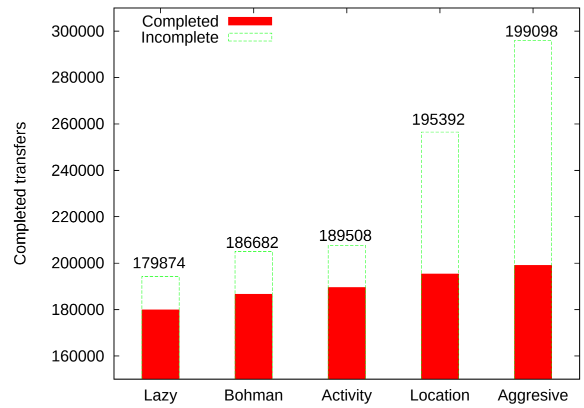


Fig. 11. Throughput of the five schemes

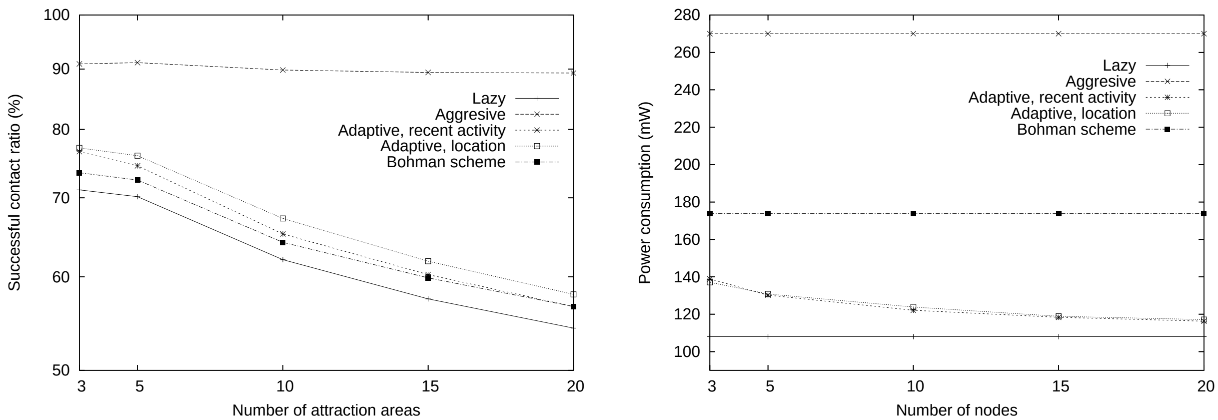
We modify the simulation so that at each encounter between two nodes, after they discover each other, they transfer a file of 100 kB in size at the typical Bluetooth transfer rate of 40 kB/s (yielding a transfer time of 2.5 s). A transfer successfully completes if the two nodes do not walk out of range sooner than 2.5 s after discovery. We run the simulation with 50 nodes and 10 attraction areas and count the number of completed and incomplete transfers. Our results are shown in Figure 11.

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 the previous subsection, is close to the energy-efficient scheme. We can see that file transfers alleviate the differences in performance results of different schemes: there is no great difference between the number of successful transfers for the lazy scheme and the aggressive one. Thus, there is no need of being more aggressive to discover more devices, because there will be no enough time to transfer data to all discovered devices. Moreover, by being less aggressive, a node may save more energy during the discovery process.

VI. RELATED WORK

Our work builds on previous research in many different fields such as opportunistic networking, epidemic algorithms for data propagation, mobility simulations in general and of ad-hoc networks in particular, and most importantly on previous work on implementing and evaluating the neighbor discovery in Bluetooth and optimizing energy efficiency of the neighbor discovery process in ad-hoc networks. In the following, we briefly describe the most relevant work in each area.

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



(a) Contacts ratio

(b) Power consumption

Fig. 9. Performance of the five schemes versus number of attraction areas

have been proposed to describe this type of communication: opportunistic networking [1], Pocket Switched Networks (PSN) [2], [3], Intermittently Connected Mobile Ad Hoc Networks (ICMAN) [13]. Opportunistic networking is considered to fall under the larger category of Delay Tolerant Networking (DTN) [17],i.e., networking in the absence of simultaneous 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 [18] or store-and-haul forwarding [19]. Several schemes have been proposed for data forwarding. Vahdat *et al.* [20] propose epidemic propagation: a flooding-style algorithm in which nodes forward messages to every node they meet. In Message Ferrying [21], 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 [22], [23] and SWIM [24]. These papers focus on routing and forwarding mechanisms and do not address the question of neighbor discovery explicitly, however neighbor discovery is a key step in establishing opportunistic communication.

User studies of mobility patterns. Murphy *et al.* [25] 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 neighbor discovery. Rather, the measurements of discovery time were done in an asymmetric fashion with one device in the inquiry mode and the other in the scan mode. Su *et al.* [1] gathered traces of human mobility patterns during two 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 neighbor discovery is performed. The same limitation applies to two other studies [2], [3] that gather data on contact frequency and duration between users carrying iMotes equipped with Bluetooth. The granularity at which the neighbor discovery process is run, i.e., every 2 minutes, implies that most brief contacts are not logged.

Bluetooth asymmetric discovery. The Bluetooth discovery procedure based on alternating the inquiry and scan modes has been extensively studied [8], [26], [27], [28], [29]. Kasten *et al.* [27] perform measurements of the discovery time for Bluetooth v1.1 devices. Other authors [26], [28] derive analytical formulae for the mean discovery time between two devices and validate them through simulation. All of these papers consider only the standard asymmetric scenario in which one of the devices is listening and the other is sending beacons.

Bluetooth symmetric neighbor discovery. Alonso *et al.* [30] study the generic neighbor discovery problem in ad-hoc networks analytically. For Bluetooth, several authors [31], [32], [33] propose and analyze symmetric neighbor discovery schemes that involve modifying the standard Bluetooth protocol [11]. By contrast, our basic discovery scheme works on currently available hardware.

Two studies [12], [7] have considered neighbor discovery in the unmodified Bluetooth protocol. As mentioned previously, Salonidis *et al.* [12] have considered a symmetric discovery scheme for Bluetooth v1.1. They were the first authors to point out the need for alternating between the inquiry and scan modes in corresponding inquire/scan phases in a peer-to-peer setting and they have also shown that the time spent in 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 [7], [27], the estimated discovery time could not be achieved experimentally. Salonidis *et al.* [12] 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. As presented previously, Bohman *et al.* [7] have further studied the Bluetooth discovery scheme and concluded that the time necessary for establishing communication using symmetric discovery is at least 8 seconds, when the mean residence time in each role is around 2.5 seconds.

Energy efficiency. Galluccio *et al.* [34] have proposed an analytical framework for studying the tradeoff between energy efficiency and time for the neighbor discovery process in ad-hoc networks. Sedov *et al*. discuss an approach for making the service discovery protocol of Bluetooth more energy-efficient [35]. Our work builds on this existing work in the area of improving energy efficiency in ad-hoc networks by introducing and evaluating two adaptive algorithms for saving energy depending on a mobility context.

VII. CONCLUSIONS

In this paper, we have presented two adaptive policies for neighbor communication establishment in Bluetooth ad-hoc networks. We adapt the power usage of the neighbor discovery process according to the probability of neighbor discovery success. Our algorithms adaptively switch between energy economic discovery modes and those that have fast discovery times in order to maximize battery life and the chances of discovering neighboring devices at the same time. Our schemes use the recent level of activity and the location of previous encounters, respectively, to predict the probability of encountering a nearby device.

In our evaluation, we first explore the energy consumption versus neighbor discovery speed trade-off in Bluetooth 1.2, select a set of representative parameter settings, then we evaluate the two adaptive schemes through a node mobility simulation. To tune the parameter settings of the neighbor discovery scheme, we develop a simulator closely matching the implementation of real devices. We use the simulator to explore a large Bluetooth parameter space and we validate its results by measurements on an experimental platform based on Cellink BTA-3100 devices compatible with Bluetooth v1.2. Our systematic search of the multi-dimensional Bluetooth parameter space provides fast and energy-efficient discovery modes. An example of a representative result, verified experimentally on our devices, is that for an average discovery interval of 1 second, which is well within the usual window of opportunity for establishing contact, the power consumption can be as low as 1.5 times the idle mode power consumption.

Our evaluation of the two adaptive schemes shows that they 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 either the duration of the experiment (because the devices drained their batteries quickly) or could not track brief contacts (due to running the discovery process infrequently on the devices). Our work provides fast and energy-efficient discovery modes, so that even short contacts can be logged, without excessive power consumption. Finally, our algorithms and techniques work on the unmodified Bluetooth standard, hence they can be readily implemented on currently deployed devices.

REFERENCES

[1] 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.

[2] 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,” Tech. Rep. UCAM-CL-TR-617, University of Cambridge, Computer Lab, 2005.

[3] 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)*, Philadelphia, PA, Aug 2005, pp. 244–251.

[4] 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)*, 2005, pp. 1–5.

[5] E. Ferro and F. Potorti, “Bluetooth and Wi-Fi Wireless Protocols: A Survey and A Comparison,” *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, Feb 2005.

[6] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, “Distributed Topology Construction of Bluetooth Personal Area Networks,” in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2001*, Anchorage, AL, USA, 2001.

[7] D. Bohman, M. Frank, P. Martini, and Scholz C., “Performance of Symmetric Neighbor Discovery in Bluetooth Ad Hoc Networks,” in *GI Jahrestagung*, 2004, vol. 1, pp. 138–142.

[8] 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.

[9] 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.

[10] S. Jain, K. Fall, and R. Patra, “Routing in a Delay Tolerant Network,” in *Proceedings of ACM SIGCOMM*, 2004.

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

[12] 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.

[13] 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.

[14] D.N. Alparslan and K. Sohraby, “A generalized random mobility model for wireless ad hoc networks and its analysis: One-dimensional case,” *IEEE/ACM Transactions on Networking*, vol. accepted for publication, 2006.

[15] C. Tuduce and T. Gross, “A Mobility Model Based on WLAN Traces and its Validation,” in *Proceedings of IEEE International Conference on Computer Communications, INFOCOM 05*, 2005.

[16] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri, “Real-world environment models for mobile network evaluation,” *IEEE Journal on Selected Areas in Communication*, vol. 23, no. 3, pp. 622–632.

[17] 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.

[18] 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.

[19] 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.

[20] A. Vahdat and D. Becker, “Epidemic Routing for Partially Connected Ad Hoc Networks,” Tech. Rep. CS-200006, Duke University, Apr 2000.

[21] 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*. 2004, ACM Press.

[22] 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)*, San Jose, CA, Oct 2002, pp. 96–107.

[23] 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)*, Boston, MA, Jun 2004, pp. 256–269.

[24] 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)," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '03)*, Annapolis, MD, 2003, pp. 233–244.

[25] 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*, Birmingham, AL, May 2002, vol. 1, pp. 414–418.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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 (ADHOCNOW'03)*, Montreal, Canada, Oct 2003, pp. 104–115.

[31] 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.

[32] 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.

[33] F. Siegemund and M. Rohs, "Rendezvous Layer Protocols for Bluetooth-Enabled Smart Devices," in *Proceedings of the International Conference on Architecture of Computing Systems (ARCS'02)*, Karlsruhe, Germany, 2002, pp. 256–273.

[34] 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*, vol. 22, no. 7, pp. 1167–1182.

[35] 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.

Catalin Drula graduated with a Master of Science degree from the Department of Computer Science of the University of Toronto in 2005. His research interests are in all aspects of mobile and ad-hoc networks. In particular, he has focused on finding new ways for disseminating information between mobile devices in ad-hoc networks. He currently works for Personal Networks, a startup company in Bucharest, Romania that he co-founded. The company develops a mobile marketing platform based on Bluetooth technology.

Cristiana Amza is an Assistant Professor with the Department of Electrical and Computer Engineering at the University of Toronto. She received her Ph.D. degree from Rice University in 2003. Her research interests are in distributed systems and Information Technology. Her current research project, Chameleon, studies infrastructure design for dynamic content dissemination with automatic adaptation to a changing environment and workload through self-managing, self-tuning and self-healing techniques. The system components are low-cost commodity components ranging from servers to desktop PC's to mobile devices.

Franck Rousseau is an Assistant Professor at INPG-Ensimag (École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble). He is a member of the LSR (JRU INPG-UJF-CNRS) Laboratory in Grenoble. He received his PhD from the INPG in 1998. Previously, he was a member of technical staff at the Open Group Research Institute in Grenoble. Dr. Rousseau's current research interests include wireless mobile networks, quality of service, ad-hoc networks, and simulation of networks.

Andrzej Duda is a Professor at the INPG-Ensimag (École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble). He received his PhD from the Université de Paris-Sud in 1984 and his Habilitation diploma from the Grenoble University in 1994. Previously, he was an Assistant Professor at the Université de Paris-Sud, a Chargé de Recherche at CNRS, a Visiting Scientist at the MIT Laboratory for Computer Science and an invited Professor at EPFL. His research interests include performance evaluation, distributed systems, and networks. He has been on program committee of several international conferences and acts as an expert for the European Commission.