Mobile Networks and Applications 10, 133–149, 2005

Efficient and Robust Protocols for Local Detection and Propagation in Smart Dust Networks *,**

IOANNIS CHATZIGIANNAKIS, SOTIRIS NIKOLETSEAS and PAUL SPIRAKIS
Computer Technology Institute (CTI) and Patras University, P.O. Box 1122, 261 10 Patras, Greece

Abstract. Smart Dust is a set of a vast number of ultra-small fully autonomous computing and communication devices, with very restricted energy and computing capabilities, that co-operate to quickly and efficiently accomplish a large sensing task. Smart Dust can be very useful in practice, i.e., in the local detection of a remote crucial event and the propagation of data reporting its realization. In this work we make an effort towards the research on smart dust from an algorithmic point of view. We first provide a simple but realistic model for smart dust and present an interesting problem, which is how to propagate efficiently information on an event detected locally. Then we present various smart dust protocols for local detection and propagation that are simple enough to be implemented on real smart dust systems, and perform, under some simplifying assumptions, a rigorous average case analysis of their efficiency and energy consumption (and their interplay). This analysis leads to concrete results showing that our protocols are very efficient and robust. We also validate the analytical results by extensive experiments.

Keywords: wireless sensor networks, algorithms, data propagation, stochastic processes, simulation

1. Introduction

Networked sensors (or Smart Dust) are very large systems, comprised of a vast number of homogenous ultra-small fully autonomous computing and communication devices that co-operate to achieve a large sensing task. Each device has one or more sensors, embedded processors and low-power radios, and is normally battery operated. Examining each such single device individually, might appear to have small utility. The realization of Smart Dust, however, lies in using and co-coordinating a vast number of such devices.

Smart Dust is a useful case of dynamic environments of networked sensors that are spread over a global system and try to communicate and compute efficiently and quickly, having only partial knowledge of the global conditions and having poor energy and computing resources. Typically, these networked sensors coordinate to perform a common task. Designing protocols to coordinate such systems (i.e., create a dynamic and efficient network of these sensors) and monitoring their behavior as they operate in complex and dynamic global environments is of great importance for information gathering and processing in many practical situations.

As an example, [11] points that integrated low-power sensing devices will permit remote object monitoring and tracking in inhospitable physical environments such as remote geographic regions or toxic urban locations. They will also enable low maintenance sensing in the field (vehicles, equipment, personnel), the office buildings (projectors, furniture, books, people), the factory floor (motors, small robotic devices).

There are many possible models for such networked sensors. In this work, we consider networked sensors where (a) all nodes in the network are homogenous and constrained by low availability of resources (energy, communication) and (b) the data being sensed by the nodes must be transmitted to *a fixed control center located far away from the sensors*. Thus direct communication between the sensor nodes and the control center is impossible and/or expensive, since there are no "high-energy" nodes through which communication can proceed. This is the general framework for MIT's μAMPS project [20], which focuses on innovative energy-optimized solutions at all levels of the system hierarchy, from the physical layer and communication protocols up to the application layer.

To motivate the challenges in designing such sensor networks, we can consider the following scenario where local detection and fast propagation to the authorities of the realization of a crucial event can be achieved using smart dust. Think about thousand of disposable sensors scattered (e.g., thrown from an aircraft) over a forest. Each of these sensors can monitor the temperature at a single, very small geographical area. The sensors coordinate to establish an efficient, dynamic and short-lived communication network, dividing the task of monitoring the terrain and offering continuous monitoring of the environment in order to alert the authorities as soon as possible after a forest fire is detected by some sensor.

Several aspects of such systems of autonomous networked entities emerge, which are quite different from those posed by standard computer networks. Such aspects include the very poor and highly restricted resources (e.g., very low battery power, low computing capabilities, total absence of synchrony and anonymity). Network protocols must be designed

* Preliminary versions of this work have appeared in the 2nd ACM Workshop on Principles of Mobile Computing (POMC, 2002 [9]) and the 3rd Workshop on Mobile and Ad-hoc Networks (WMAN, 2003 [6]).

** This work has been partially supported by the IST Program of the European Union under contract numbers IST-1999-14186 (ALCOM-FT) and IST-2001-33116 (FLAGS).

to achieve fault tolerance in the presence of individual node failure while minimizing energy consumption. Another important aspect is the scalability to the change in network size, node density and topology. The network topology changes over time as some nodes may die, or possibly because new nodes join later.

This work, continuing our line of research on communication in ad-hoc mobile networks [7,8], is an attempt towards capturing the underlying foundational and algorithmic issues in the design of such systems, abstracting accurately enough the real technological specifications involved and providing some first concrete results for the efficiency of a variety of smart dust protocols using an average case analysis. We focus in this paper on the efficient use of smart dust in local detection and propagation protocols. We first provide an abstract model for smart dust systems which is simple enough to allow an analysis to develop, being however at the same time quite realistic, in terms of the technological specifications of real smart dust systems it captures. Then we define the problem of local detection and propagation using smart dust and also propose some concrete performance and robustness measures for the average case analysis of protocols for this problem.

Our results

For the local detection and propagation problem using smart dust, we provide three protocols. All protocols are simple enough to be implemented in real smart dust systems despite the severe energy and computing power limitations of such systems. Furthermore, we give a rigorous average case analysis for the efficiency of these protocols. We consider a variety of performance and robustness criteria, such as propagation time, number of particle to particle transmissions (which also characterizes energy consumption and time efficiency, assuming an efficient MAC protocol) and fault-tolerance:

1. Our first protocol, which we call the "local target protocol" (LTP), uses a fast and cheap search phase which is assumed to always return a nearby particle towards the authorities, uniformly in some range. We show that LTP is efficient, in the sense that it achieves a propagation time and an energy consumption whose expected ratio over the optimal solutions is at most $\pi/2 \approx 1.57$.
2. Our second protocol, the "min-two uniform targets" protocol (m2TP), applies the simple idea of getting at least two particles towards the authorities and selecting the best in terms of propagation progress. It is, in fact, an optimized and more efficient version of the local target protocol, and has an expected time and energy ratio over the optimal solutions which is at most $\pi^2/8 \approx 1.24$.
3. Next we provide tight upper bounds to the distribution of the number of particle to particle data transmissions (and thus the efficiency) of a generalized target protocol.
4. We propose a new protocol which we call the "Sleep–Awake" protocol (SWP), that *explicitly* uses the energy saving characteristics, such as the alteration of sleep and awake time periods, of smart dust particles. By using both analytic and extensive experimental means, we investigate the relation between (a) the success probability and (b) the time efficiency of the protocol, to the maximum sleeping time period, for various values of other parameters, such as particle density, particle distribution and angle α. We interestingly note that the new protocol is efficient, despite the fact that the particles are allowed to enter a sleeping mode in order to save energy.

All protocols mentioned above are shown to be robust in the following sense: (a) the protocols use the search and the backtrack phases to explore the active (non-faulty) "next" particles. Thus, the fact (demonstrated both by analysis and simulation) that the protocols succeed with high probability, exhibits also fault-tolerance properties of the protocols; (b) our findings showing that the protocols succeed even in the case of low densities also implies robustness.

Discussion of selected related work

In the last few years, Sensor Networks have attracted a lot of attention from researchers at all levels of the system hierarchy, from the physical layer and communication protocols up to the application layer.

At the MAC level, many researchers have done research work in an effort to minimize the power consumption. [27] presents a contention-based protocol that tries to minimize energy consumption due to node idle listening, by avoiding the overhearing among neighboring nodes. A recent work [30] exploits a similar method for energy savings, and further reduce idle listening by avoiding any use of out-of-channel signaling. Additionally, their protocol trades off per-node fairness for further energy savings.

For establishing communication and routing information to the control center, mobile ad-hoc routing protocols [24] may be used in sensor networks. However, although protocols for mobile ad-hoc networks take into consideration energy conservation issues, most of them are not really suitable for sensor networks. [19] presents a routing protocol suitable for sensor networks that makes *greedy* forwarding decisions using only information about a node's immediate neighbors in the network topology. This approach achieves high scalability as the density of the network increases. [14] presents a clustering-based protocol that utilizes randomized rotation of local cluster heads to evenly distribute the energy load among the sensors in the network. [21] introduces a new energy efficient routing protocol that does not provide periodic data monitoring (as in [14]), but instead nodes transmit data only when sudden and drastic changes are sensed by the nodes. As such, this protocol is well suited for time critical applications and compared to [14] achieves less energy consumption and response time.

A family of negotiation-based information dissemination protocols suitable for wireless sensor networks is presented in [15]. Sensor Protocols for Information via Negotiation (SPIN) focus on the efficient dissemination of individual sensor observations to all the sensors in a network. However, in contrast to classic flooding, in SPIN sensors negotiate with each other about the data they possess using meta-data names. These negotiations ensure that nodes only transmit data when necessary, reducing the energy consumption for useless transmissions.

A data dissemination paradigm called *directed diffusion* for sensor networks is presented in [18], where data-generated by sensor nodes is named by attribute–value pairs. An observer requests data by sending *interests* for named data; data matching the interest is then "drawn" down towards that node by selecting a single path or through multiple paths by using a low-latency tree. [17] presents an alternative approach that constructs a greedy incremental tree that is more energy-efficient and improves path sharing.

We note that, as opposed to the work presented in this paper, the above research focuses on energy consumption without examining the time efficiency of their protocols. Furthermore, these works contain basically protocol design and technical specifications, while quantitative aspects are only experimentally evaluated and no theoretical analysis is given. Note also that our protocols are quite general in the sense that (a) do not assume global network topology information, (b) do not assume geolocation information (such as GPS information) and (c) use very limited control message exchanges, thus having low communication overhead.

Finally, our third protocol is using a similar approach to the recent work of [26], where a new technique called Sparse Topology and Energy Management (STEM) is proposed that aggressively puts nodes to sleep. Interestingly, the analysis and experiments of STEM show improvements of nearly two orders of magnitude compared to sensor networks without topology management.

Some recent work

In [4] the authors present a new protocol for data propagation that avoids flooding by probabilistically favoring certain ("close to optimal") data transmissions. As shown by a geometry analysis, the protocol is *correct*, since it always propagates data to the sink, under ideal network conditions (no failures). Using stochastic processes, they show that the protocol is *very energy efficient*. Also, when part of the network is inoperative, the protocol manages to propagate data very close to the sink, thus in this sense it is *robust*. They finally present and discuss large-scale experimental findings validating the analytical results.

In [5], the authors have implemented and experimentally evaluated two variations of LTP, under new, more general and realistic modelling assumptions. They comparatively study LTP to PFR, by using extensive experiments, highlighting their relative advantages and disadvantages. All protocols are very successful. In the setting considered there, PFR seems to be faster while the LTP based protocols are more energy efficient.

In [12], Euthimiou et al. study the problem of *energy-balanced* data propagation in wireless sensor networks. The energy balance property guarantees that the average per sensor energy dissipation is the same for all sensors in the network, during the entire execution of the data propagation protocol. This property is important since it prolongs the network's lifetime by avoiding early energy depletion of sensors. They propose a *new algorithm* that in each step decides whether to propagate data one-hop towards the final destination (the sink), or to send data directly to the sink. This randomized choice balances the (cheap) one-hop transimssions with the direct transimissions to the sink, which are more expensive but "*bypass*" *the sensors lying close to the sink*. Note that, in most protocols, these close to the sink sensors tend to be overused and die out early.

In [1], the authors propose a new energy efficient and fault tolerant protocol for data propagation in smart dust networks, the Variable Transmission Range Protocol (VTRP). The basic idea of data propagation in VTRP is the varying range of data transmissions, i.e., they allow the transmission range to increase in various ways. Thus data propagation in the protocol exhibits high fault-tolerance (by bypassing obstacles or faulty sensors) and increases network lifetime (since critical sensors, i.e., close to the control center are not overused). They *implement* the protocol and perform an *extensive experimental evaluation and comparison to a representative protocol* (LTP) of several important performance measures with a focus on energy consumption. The findings indeed demonstrate that the protocol achieves significant improvements in energy efficiency and network lifetime.

In [23], Nikoletseas et al. (a) propose *extended versions* of two data propagation protocols: the Sleep–Awake Probabilistic Forwarding Protocol (SW-PFR) and the Hierarchical Threshold sensitive Energy Efficient Network protocol (H-TEEN). These non-trivial extensions aim at improving the performance of the original protocols, by introducing *sleep–awake periods* in the PFR protocol to save energy, and introducing a *hierarchy of clustering* in the TEEN protocol to better cope with large networks areas; (b) they have *implemented* the two protocols and performed an *extensive experimental comparison* (using simulation) of various important measures of their performance with a focus on energy consumption; (c) they investigate in detail the *relative advantages and disadvantages* of each protocol and discuss and explain their behavior; (d) in the light above they propose and discuss a possible *hybrid combination of the two protocols* towards optimizing certain goals. Efficient collision avoidance protocols, particularly useful for multipath data propagation, have been proposed in [10].

A brief description of the technical specifications of state-of-the-art sensor devices, a discussion of possible models used to abstract such networks and a presentation of some characteristic protocols for data propagation in sensor networks, along with an evaluation of their performance analysis, can be found in the recent book chapter of Boukerche and Nikoletseas [3].

2. The model

Smart dust is comprised of a vast number of ultra-small homogenous sensors, which we call "grain" particles. Each

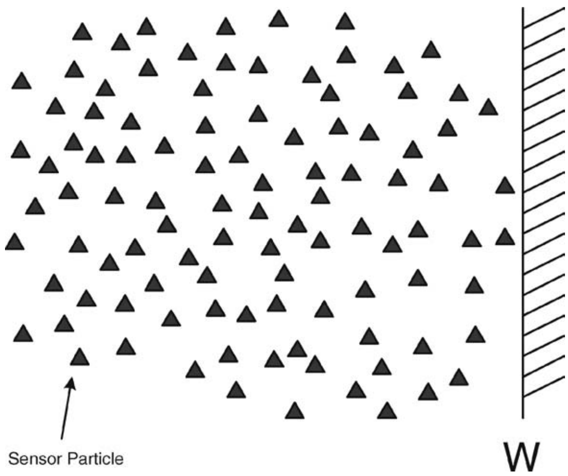

Figure 1. A smart dust cloud.

smart-dust *grain particle* is a fully-autonomous computing and communication device, characterized mainly by its available power supply (battery) and the energy cost of computation and transmission of data. Such particles cannot move.

Each particle is equipped with a set of monitors (sensors) for light, pressure, humidity, temperature, etc. Each particle has two communication modes: a *broadcast* (digital radio) *beacon mode* (for low energy – short signals) and a *directed to a point* actual data transmission mode (usually via a laser beam). Also, in a variation of our model capturing energy saving specifications, each particle may alternate (independently of other particles) between a *sleeping* and an *awake* mode. During sleeping periods grain particles cease any communication with the environment, thus they are unable to listen, receive and propagate data transmitted by other particles. Depending on the specific application the sensing part may cease or not during the sleeping mode. In the case where sensing is not ceased during sleeping mode, detection of the crucial event wakes the particle up.

We adopt here (as a starting point) a two-dimensional (plane) framework: a *smart dust cloud* (a set of particles) is spread in an area (for a graphical presentation, see figure 1). Note that a two-dimensional setting is also used in [14,15,17,18,21].

Definition 2.1. Let d (usually measured in numbers of particles/m^2) be the *density* of particles in the area. Let $\mathcal{R}$ be the maximum (beacon/laser) transmission range of each grain particle. A *receiving wall* $\mathcal{W}$ is defined to be an infinite line in the smart-dust plane. Any particle transmission within range $\mathcal{R}$ from the wall $\mathcal{W}$ is received by $\mathcal{W}$.

We assume that $\mathcal{W}$ has very strong computing power, able to collect and analyze received data and has a constant power supply and so has no energy constraints. The wall represents, in fact, the authorities (the fixed control center) who the realization of a crucial event should be reported to. Note that a wall of appropriately big (finite) length suffices. We plan to conduct an analysis of the (expected and/or with high probability) deviation of the transmitted data from the vertical to the wall position in order to provide upper bounds on the wall's length needed.

Furthermore, we assume that there is a set-up phase of the smart dust network, during which the smart cloud is dropped in the terrain of interest, when using special control messages (which are very short, cheap and transmitted only once) each smart dust particle is provided with the direction of $\mathcal{W}$. By assuming that each smart-dust particle has individually *a sense of direction* (e.g., through its magnetometer sensor), and using these control messages, each particle is aware of the general location of $\mathcal{W}$.

We feel that our model, although simple, depicts accurately enough the technological specifications of real smart dust systems. Similar models are being used by other researchers in order to study sensor networks (see [14,21]). In contrast to [18,19], our model is weaker in the sense that *no geolocation abilities* are assumed (e.g., a GPS device) for the smart dust particles leading to more generic and thus stronger results. In [16] a thorough comparative study and description of smart dust systems is given, from the technological point of view. In the following section we report some basic technical characteristics which we took into account when defining the model of smart dust we use here.

3. Technological specifications of smart dust devices

New technology is changing the nature of sensors and the way they interface with data acquisition and control systems. Researchers have developed an open-source hardware and software platform that combines sensing, communications, and computing into a complete architecture. The first commercial generation of this platform was dubbed the Rene Mote, and several thousand of these sensors have been deployed at commercial and research institutions worldwide to promote the development and application of wireless sensor networks.

The platforms development community is based on the open-source model, which has become well known with the increasingly popular Linux operating system. Most development work is done in the public domain, and it includes the hardware design and software source code. Users of the technology contribute their developments back to the community so that the base of code and hardware design grows rapidly.

It is worth noting that currently, a number of research institutions in the U.S. are working on centimeter-scale (and even smaller) distributed sensor networks [2,29].

3.1. Hardware design of wireless sensors

The basic MICA hardware uses a fraction of a Watt of power and consists of commercial components a square inch in size. The hardware design consists of a small, low-power radio and processor board (known as a mote processor/radio, or MPR, board) and one or more sensor boards (known as a mote sensor, or MTS, board). The combination of the two types of boards form a networkable wireless sensor.

The MPR board includes a processor, radio, A/D converter, and battery. The processor is an ATMEL ATMEGA, but there

CPU speed	4 MHz
Memory	ROM: 128 Kb FLASH SDRAM: 4 Kb EEPROM: 4 Kb
Power supply	2 AA batteries
Power consumption	0.75 mW
Processor current draw	5.5 mA (active current) < 20 μA (sleep mode)
Radio current draw	12 mA (transmit current) 1.8 mA (receive current) < 1 μA (sleep mode)
Output device	3 LEDs
I/O port	Expansion connected (51 pin) Serial port (proprietary 16-pin)
Network	Wireless 4 Kbits/s at 916 MHz (ISM band) Radio range depends on antennae configuration

Figure 2. MPR300CB specifications.

are other processors that would meet the power and cost targets. The processor runs at 4 MHz, has 128 Kb of flash memory and 4 Kb of SDRAM. In a given network, thousands of sensors could be continuously reporting data, creating heavy data flow. Thus, the overall system is memory constrained, but this characteristic is a common design challenge in any wireless sensor network.

The MPR modules contain various sensor interfaces, which are available through a small 51-pin connector that links the MPR and MTS modules. The interface includes: an 8-channel, 10-bit A/D converter; a serial UART port; and an I2C serial port. This allows the MPR module to connect to a variety of MTS sensor modules, including MTS modules that use analog sensors as well as digital smart sensors. The MPR module has a guaranteed unique, hard-coded 64-bit address.

The processors, radio, and a typical sensor load consumes about 100 mW in active mode. This figure should be compared with the 30 μA draw when all components are in sleep mode. Figure 2 shows a synopsis of the MPR specs.

The MTS sensor boards currently include light/temperature, two-axis acceleration, and magnetic sensors and 420 mA transmitters. The wireless transmission is at 4 Kbps rate and the transmission range may vary. Researchers are also developing a GPS board and a multisensor board that incorporates a small speaker and light, temperature, magnetic, acceleration, and acoustic (microphone) sensing devices. The MICA developers community welcomes additional sensor board designs.

3.2. Software and the TinyOS

A considerable portion of the challenge faced by the developers of MICA devices is in the software embedded in the sensors. The software runs the hardware and networkmaking sensor measurements, routing measurement data, and controlling power dissipation. In effect, it is the key ingredient that makes the wireless sensor network produce useful information.

To this end, a lot of effort has gone into the design of a software environment that supports wireless sensors. The result is a very small operating system named TinyOS, or Tiny Microthreading Operating System, which allows the networking, power management, and sensor measurement details to be abstracted from the core application development. The operating system also creates a standard method of developing applications and extending the hardware. Although tiny, this operating system is quite efficient, as shown by the small stack handling time. Figure 3 lists the key points of TinyOS.

Software footprint	3.4 Kb
Transmission cost	1 μJ/bit
Inactive state	< 25 μA
Peak load	20 mA
Typical CPU usage	$< 50\%$
Events propagate thru stack	< 40 μs

Figure 3. TinyOS key facts.

4. The problem

An *adversary* $\mathcal{A}$ selects a single particle, p, in the plane-cloud and allows it to monitor a *local crucial event* $\mathcal{E}$. The general *propagation problem* $\mathcal{P}$ is the following:

> "How can particle p, via cooperation with the rest of the cloud, propagate information about event $\mathcal{E}$ to the receiving wall $\mathcal{W}$"?

Definition 4.1. Let $h_{\text{opt}}(p, \mathcal{W})$ be the (optimal) number of "hops" (direct, vertical to $\mathcal{W}$ transmissions) needed to reach the wall, in the *ideal* case in which particles always exist in pair-wise distances $\mathcal{R}$ in the vertical line from p to $\mathcal{W}$. Let Π be *a smart-dust propagation protocol*, using *a transmission path* of length $L(\Pi, p, \mathcal{W})$ to send info about event $\mathcal{E}$ to wall $\mathcal{W}$. Let $h(\Pi, p, \mathcal{W})$ be the number of hops (transmissions) taken to reach $\mathcal{W}$. The "*hops*" *efficiency* of protocol Π is the ratio

$$C_h = \frac{h(\Pi, p, \mathcal{W})}{h_{\text{opt}}(p, \mathcal{W})}.$$

Clearly, the number of hops (transmissions) needed characterizes the energy consumption and the time needed to propagate the information $\mathcal{E}$ to the wall. Remark that $h_{\text{opt}} = \lceil d(p, \mathcal{W})/\mathcal{R} \rceil$, where $d(p, \mathcal{W})$ is the (vertical) distance of p from the wall $\mathcal{W}$.

In the case where protocol Π is randomized, or in the case where the distribution of the particles in the cloud is a random distribution, the number of hops h and the efficiency ratio C_h are random variables and we study here their expected values.

The reason behind these definitions is that when p (or any intermediate particle in the information propagation to $\mathcal{W}$) "looks around" for a particle as near to $\mathcal{W}$ as possible to pass its information about $\mathcal{E}$, it may not get any particle in the perfect direction of the line vertical to $\mathcal{W}$ passing from p. This difficulty comes mainly from three causes: (a) due to the initial spreading of particles of the cloud in the area and because particles do not move, there might not be any particle in that direction; (b) particles of *sufficient remaining* battery power may not be available in the right direction; (c) particles may temporarily "sleep" (i.e., not listen to transmissions) in order to save battery power.

Remark. Note that any given distribution of particles in the smart dust cloud may not allow the ideal optimal number of hops to be achieved at all. In fact, the least possible number of hops depends on the input (the positions of the grain particles). We have chosen, however, to compare the efficiency of our protocols to the ideal case. A comparison with the best achievable number of hops in each input case will of course give better efficiency ratios for our protocols.

5. The local target protocol (LTP)

Let $d(p_i,\ p_j)$ the distance (along the corresponding vertical lines towards $\mathcal{W}$) of particles $p_i,\ p_j$ and $d(p_i, \mathcal{W})$ the (vertical) distance of p_i from $\mathcal{W}$. Let *info*($\mathcal{E}$) the information about the realization of the crucial event $\mathcal{E}$ to be propagated. In this protocol, each particle p' that has received *info*($\mathcal{E}$) from p (via, possibly, other particles) does the following:

- *Search phase*. It uses a periodic low energy broadcast of a beacon in order to discover a particle nearer to $\mathcal{W}$ than itself (i.e., a particle p'' where $d(p'', \mathcal{W}) < d(p', \mathcal{W})$).
- *Direct transmission phase*. Then, p' sends *info*($\mathcal{E}$) to p'' via a direct line (laser) time consuming transmission.
- *Backtrack phase*. If consecutive repetitions of the *search phase* fail to discover a particle nearer to $\mathcal{W}$, then p' sends *info*($\mathcal{E}$) to p (i.e., to the particle that it originally received the information).

Note that one can estimate an a-priori upper bound on the number of repetition of the search phase needed, by using the probability of success of each search phase. This bound can be used to decide when to backtrack.

Also note that the maximum distance $d(p',\ p'')$ is $\mathcal{R}$, i.e., the beacon transmission range (for a graphical representation see figures 4, 5).

To enable a first step towards a rigorous analysis of smart dust protocols, we make the following simplifying assumption. *The search phase takes zero time* and always finds a p'' (of sufficiently high battery) in the semicircle of center p, in the direction towards $\mathcal{W}$. Note that this assumption on always finding a particle can be relaxed in the following ways: (a) by repetitions of the search phase until a particle is found. This makes sense if at least one particle exists but was sleeping during the failed searches; (b) we may consider, instead of just the semicircle, a cyclic sector defined by circles of radiuses $\mathcal{R} - \Delta\mathcal{R},\ \mathcal{R}$ and also take into account the density of the smart cloud; (c) if the protocol during a search phase ultimately fails to find a particle towards the wall, it may *backtrack*.

In this analysis we do not consider the energy spent in the search phase. Note, however, that even the case where this is comparable to the energy spent in actual data transmission, the number of hops accounts for both (total energy spent is upper bounded by a multiple of actual data transmission energy).

We also assume that the position of p'' is uniform in the arc of angle $2a$ around the direct line from p' vertical to $\mathcal{W}$. Each data transmission (one hop) takes constant time t (so the "hops" and time efficiency of our protocols coincide in this case). We also assume that each target selection is random *independent* of the others, in the sense that it is always drawn uniformly in the arc $(-\alpha, \alpha)$.

We are aware of the fact that the above assumptions may not be very realistic in practice, however, they allows us to perform a first effort towards providing some concrete analytical results.

Lemma 5.1. The expected "hops" efficiency of the local target protocol in the α-uniform case is $E(C_h) \simeq \alpha / \sin\alpha$, for large h_{opt}. Also $1 \leqslant E(C_h) \leqslant \pi/2 \approx 1.57$, for $0 \leqslant \alpha \leqslant \pi/2$.

Proof. Due to the protocol, a sequence of points is generated, $p_0 = p, p_1, p_2, \ldots, p_{h-1}, p_h$ where p_{h-1} is a particle within $\mathcal{W}$'s range and p_h is part of the wall. Let α_i be the (positive or negative) angle of p_i with respect to p_{i-1}'s verti-

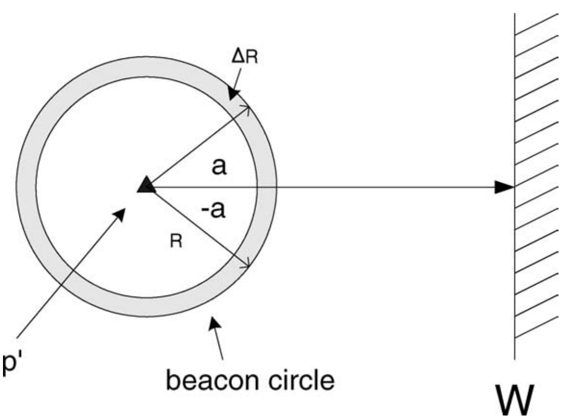

Figure 4. Example of the search phase.

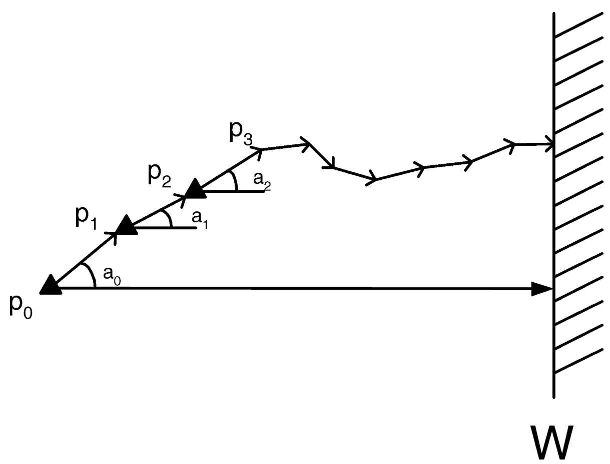

Figure 5. Example of a transmission.

cal line to $\mathcal{W}$. It is:

$$\sum_{i=1}^{h-1} d(p_{i-1}, p_i) \leqslant d(p, \mathcal{W}) \leqslant \sum_{i=1}^{h} d(p_{i-1}, p_i).$$

Since the (vertical) progress towards $\mathcal{W}$ is then $\Delta_i = d(p_{i-1}, p_i) = \mathcal{R}\cos\alpha_i$, we get:

$$\sum_{i=1}^{h-1} \cos\alpha_i \leqslant h_{\text{opt}} \leqslant \sum_{i=1}^{h} \cos\alpha_i.$$

From Wald's equation for the expectation of a sum of a random number of independent random variables (see [25]), then

$$E(h-1)E(\cos\alpha_i) \leqslant E(h_{\text{opt}}) = h_{\text{opt}} \leqslant E(h)E(\cos\alpha_i).$$

Now, $\forall i$, $E(\cos\alpha_i) = \int_{-\alpha}^{\alpha} \cos x (1/2\alpha)\,\mathrm{d}x = \sin\alpha/\alpha$. Thus

$$\frac{\alpha}{\sin\alpha} \leqslant \frac{E(h)}{h_{\text{opt}}} = E(C_h) \leqslant \frac{\alpha}{\sin\alpha} + \frac{1}{h_{\text{opt}}}.$$

Assuming large values for h_{opt} (i.e., events happening far away from the wall, which is the most interesting case in practice since the detection and propagation difficulty increases with distance) we have (since for $0 \leqslant \alpha \leqslant \pi/2$ it is $1 \leqslant \alpha/\sin\alpha \leqslant \pi/2$) we get the result. □

6. Local optimization – the "min two uniform targets" protocol (m2TP)

Note that the same basic framework holds for any situation in which the local (vertical) progress in the direction towards $\mathcal{W}$ (i.e., Δ_i) is of the same, independent, distribution. I.e., it always holds (via the Wald's equation) that

$$\begin{aligned} &\frac{\mathcal{R}}{E(\Delta_i)} \leqslant \frac{E(h)}{h_{\text{opt}}} \leqslant \frac{\mathcal{R}}{E(\Delta_i)} + \frac{1}{h_{\text{opt}}} \\ \Rightarrow\quad &E(C_h) \approx \frac{\mathcal{R}}{E(\Delta_i)} = \frac{1}{E(\cos\alpha_i)} \end{aligned} \tag{1}$$

for large h. To understand the power of this, let us assume that the search phase always returns *two points* p'', p''' each uniform in $(-\alpha, \alpha)$ and that the protocol selects the best of the two points, with respect to the local (vertical) progress.

Lemma 6.1. The expected "hops" efficiency of the "min two uniform targets" protocol in the α-uniform case is

$$E(C_h) \approx \frac{\alpha^2}{2(1-\cos\alpha)},$$

for $0 \leqslant \alpha \leqslant \pi/2$ and for large h.

Proof. Let α_{i1}, α_{i2} the angles of the particles found and let $\alpha_i = \min\{|\alpha_{i1}|, |\alpha_{i2}|\}$. Then, for any $0 \leqslant \phi \leqslant \alpha$, it is:

$$\begin{aligned} \mathbb{P}\{\alpha_i > \phi\} &= \mathbb{P}\big\{|\alpha_{i1}| > \phi \cap |\alpha_{i2}| > \phi\big\} \\ &= \left(\frac{2\alpha - 2\phi}{2\alpha}\right)^2 \\ &= \left(\frac{\alpha-\phi}{\alpha}\right)^2. \end{aligned}$$

Thus, the distribution function of α_i, for any $0 \leqslant \phi \leqslant \alpha$, is

$$F_{\alpha_i}(\phi) = \mathbb{P}\{\alpha_i \leqslant \phi\} = 1 - \frac{(\alpha-\phi)^2}{\alpha^2} = \frac{2\alpha\phi - \phi^2}{\alpha^2}$$

and the probability density function is, for any $0 \leqslant \phi \leqslant \alpha$:

$$f_{\alpha_i}(\phi) = \frac{\mathrm{d}}{\mathrm{d}\phi}\mathbb{P}\{\alpha_i \leqslant \phi\} = \frac{2}{\alpha}\left(1 - \frac{\phi}{\alpha}\right).$$

The expected local progress is:

$$E(\cos\alpha_i) = \int_0^{\alpha} \cos\phi f_{\alpha_i}(\phi)\,\mathrm{d}\phi = \frac{2(1-\cos\alpha)}{\alpha^2}. \tag{2}$$

□

We remark that

$$\lim_{\alpha\to 0} E(C_h) = \lim_{\alpha\to 0} \frac{2\alpha}{2\sin a} = 1$$

and

$$\lim_{\alpha\to\pi/2} E(C_h) = \frac{(\pi/2)^2}{2(1-0)} = \frac{\pi^2}{8} \approx 1.24.$$

Lemma 6.2. The expected "hops efficiency of the min-two uniform targets protocol is $1 \leqslant E(C_h) \leqslant \pi^2/8 \approx 1.24$ for large h and for $0 \leqslant \alpha \leqslant \pi/2$.

We remark that, w.r.t. the expected hops efficiency of the local target protocol, the min-two uniform targets protocol achieves, because of the one additional search, a relative gain which is $(\pi/2 - \pi^2/8)/(\pi/2) \approx 21.5\%$. We experimentally investigate the further gain of additional (i.e., $m > 2$) searches in section 10.

7. Tight upper bounds to the hops distribution of the general target protocol

Consider the particle p (which senses the crucial event) at distance x from the wall. Let us assume that when p searches in the sector S defined by angles $(-\alpha, \alpha)$ and radius $\mathcal{R}$, another particle p' is returned *in the sector* with some probability density $f(\overrightarrow{p'})\,\mathrm{d}\mathcal{A}$, where $\overrightarrow{p'} = (x_{p'}, y_{p'})$ is the position of p' in S and $\mathrm{d}\mathcal{A}$ is an infinitesimal area around p'.

Definition 7.1 (Horizontal progress). Let Δx be the projection of the line segment (p, p') on the line from p vertical to $\mathcal{W}$.

We assume that each search phase returns such a particle, with independent and identical distribution $f(\cdot)$.

Definition 7.2 (Probability of significant progress). Let $m > 0$ be the least integer such that $\mathbb{P}\{\Delta x > \mathcal{R}/m\} \geqslant p$, where $0 < p < 1$ is a given constant.

Lemma 7.1. For each continuous density $f(\cdot)$ on the sector S and for any constant p, there is always an $m > 0$ as above.

Proof. Remark that $f(\cdot)$ defines a density function $\widetilde{f}(\cdot)$ on $(0, \mathcal{R}]$ which is also continuous. Let $\widetilde{F}()$ its distribution function. Then we want $1-\widetilde{F}(\mathcal{R}/m) \geqslant p$, i.e., to find the first m such that $1-p \geqslant \widetilde{F}(\mathcal{R}/m)$. Such an m always exists since $\widetilde{F}$ is continuous in [0, 1]. □

Definition 7.3. Consider the (discrete) stochastic process P in which with probability p the horizontal progress is $\mathcal{R}/m$ and with probability q it is zero, where $q = 1-p$.

Let Q the actual stochastic process of the horizontal progress implied by $f(\cdot)$.

Lemma 7.2. $\mathbb{P}_P\{h \leqslant h_0\} \leqslant \mathbb{P}_Q\{h \leqslant h_0\}$.

Proof. The actual process Q makes always more progress than P. □

Now let $t = \lceil x/(\mathcal{R}/m)\rceil = \lceil mx/\mathcal{R}\rceil$. Consider the integer random variable H such that $\mathbb{P}\{H = i\} = q^i(1-q)$ for any $i \geqslant 0$. Then H is geometrically distributed. Let $H_1, \dots, H_t$ be t random variables, independent and identically distributed according to H. Clearly then

Lemma 7.3. $\mathbb{P}_P\{\text{number of hops is } h\} = \mathbb{P}\{H_1 + \cdots + H_t = h\}$.

The probability generating function of H is

$$H(s) = \mathbb{P}\{H=0\} + \mathbb{P}\{H=1\}s + \cdots + \mathbb{P}\{H=i\}s^i + \cdots,$$

i.e.,

$$H(s) = p\big(1 + qs + q^2s^2 + \cdots + q^is^i + \cdots\big) = \frac{p}{1-qs}.$$

But the probability generating function of $\sum_t = H_1 + \cdots + H_t$ is then just $(p/(1-qs))^t$ by the convolution theorem of generating functions. This is just the generating function of the t-fold convolution of geometric random variables, and it is *exactly* the distribution of the *negative binomial distribution* (see [13], vol. 1, p. 253). Thus,

Theorem 7.4.

$$\begin{aligned}\mathbb{P}_P\{\text{the number of hops is } h\} &= \binom{-t}{h} p^t(-q)^h \\ &= \binom{t+h-1}{h} p^t q^h.\end{aligned}$$

Corollary 7.5. For the process P, the mean and variance of the number of hops are:

$$E(h) = \frac{tq}{p}, \qquad Var(h) = \frac{tq}{p^2}.$$

Note that the method sketched above, finds a distribution that upper bounds the number of hops till the crucial event is reported to the wall. Since for all $f(\cdot)$ it is $h \geqslant x/\mathcal{R} = h_{\text{opt}}$ we get that

$$\frac{E_P(h)}{h_{\text{opt}}} \leqslant \frac{\lceil mx/\mathcal{R}\rceil q/p}{x/\mathcal{R}} \leqslant \frac{(m+1)q}{p}.$$

Theorem 7.6. The above upper bound process P estimates the expected number of hops to the wall with a guaranteed efficiency ratio $(m+1)/(1-p)p$ at most.

Example. When for $p = 0.5$ we have $m = 2$ and the efficiency ratio is 3, i.e., the overestimate is 3 times the optimal number of hops.

8. The "sleep–awake" protocol (SWP)

We now present a new protocol for smart dust networks which we call the "sleep–awake" protocol. In contrast to the previous protocols, we now assume that we can explicitly use the periods that a particle is in *awake mode* or in *sleeping mode*. During sleeping periods grain particles cease any communication with the environment, thus they are unable to listen, receive and propagate data transmitted by other particles.

The procedures of search transmission and backtrack are the same as in the LTP.

In the above procedure, propagation of *info*(E) is done in two steps; (i) particle p' locates the next particle (p'') and transmits the information and (ii) particle p' waits until the next particle (p'') succeeds in propagating the message further towards $\mathcal{W}$. In both steps particle p' will remain *awake*. This is done to speed up the backtrack phase in case p'' does not succeed in discovering a particle nearer to $\mathcal{W}$. Note, however, that as soon as p'' succeeds to propagate data, p' resumes its sleep–awake mode.

Propagation protocols for such energy-restricted systems should at least guarantee that the wall *eventually* receives the messages that report a crucial event. The success of such protocols depends on the density d of grain particles/m^2 and their distribution, the distribution of sleeping and awake time periods and, of course, on the angle α of the search beacon.

We below provide some first results on the interplay between these parameters. In particular, we focus on the relation between the maximum sleeping time period and the other parameters, thus allowing to program the smart-cloud energy saving specifications accordingly.

To simplify the analysis we assume that the grain particles are uniformly distributed on the smart-dust plane. Thus, in the area inspected during a search phase of beacon angle α between R and $\mathcal{R} + \Delta\mathcal{R}$, the number of grain particles is

$$\begin{aligned} N &= d\left(\frac{\alpha}{\pi}\pi\mathcal{R}^2 - \frac{\alpha}{\pi}\pi(\mathcal{R}-\Delta\mathcal{R})^2\right) \\ &\simeq \alpha\big(2\mathcal{R}\Delta\mathcal{R} - \Delta\mathcal{R}^2\big)d \simeq 2\alpha d\mathcal{R}\Delta\mathcal{R}. \end{aligned} \tag{3}$$

Now, we assume that the sleeping/awake time durations alternate independently in each particle and have lengths s, w, respectively (this can be easily achieved if during the start-up phase, the first awake period w is set using a random bit generator, or is hardcoded into the particle by the manufacturer). Thus, the probability that at least one of the N particles in the sender's beacon search area is awake is:

$$P_1 = \mathbb{P}\{\text{at least one particle is awake}\} = 1 - \left(\frac{s}{s+w}\right)^N = 1 - \left(\frac{s}{s+w}\right)^{2\alpha d \mathcal{R} \Delta \mathcal{R}}.$$

Thus the probability that the event report eventually reaches the wall is:

$$\mathbb{P}\{success\} = \sum_{h_0=1}^{\infty} P_1^h \, \mathbb{P}\{h = h_0\},$$

where $\mathbb{P}\{h = h_0\}$ is the probability density function of the random variable h.

Let now $\beta = s/w$, i.e., β represents the energy saving specifications of the smart dust particles (a typical value for β may be 100). Then,

Definition 8.1. The energy saving specification is:

$$en = \frac{s}{s+w} = 1 - \frac{1}{1+\beta}.$$

By taking d such that $2\alpha \Delta \mathcal{R} d \simeq n(1+\beta)$ we get $P_1 = 1 - \mathrm{e}^{-n}$. Then, by the Bernoulli inequality, we have $P_1^{E(h)} \geqslant 1 - E(h)\mathrm{e}^{-n}$. This probability is *non-zero* when

$$n > \ln E(h).$$

This final condition allows to set the technical specifications and the propagation time accordingly in order to guarantee that the crucial event is eventually reported to the wall.

9. Implementation aspects and details

We now proceed by providing a more detailed description of the protocols implementation in our simulation environment. We also discuss implementation aspects of our protocols in current technology wireless sensor networks.

We assume that the particles are equipped with TinyOS, an event driven operating system suitable for smart dust [28]. The pseudo-code presented in figures 6–8 demonstrates how to implement the SWP protocol. Note that the implementation of LTP is very similar.

At every particle, we use a Boolean variable `HOLDER` to denote the status of the particle. It is set to true only if the site holds *info*($\mathcal{E}$) (the information about the realization of the crucial event $\mathcal{E}$ to be propagated). A variable `PREVIOUS` records the particle from which *info*($\mathcal{E}$) was received. A set `OUTf` is used to store any particle that failed to propagate a message towards the wall (this set is used for backtracking purposes).

```
msgHandler rcvInfo(msg) {
  Timer.stop();
  HOLDER = true;
  PREVIOUS = sender(msg);
}

msgHandler rcvReqBeacon(msg) {
  initiator = sender(msg);
  send(initiator)[BeaconMsg];
}

msgHandler rcvBeacon(msg) {
  remember(sender(msg), power(msg));
}

msgHandler rcvSuccess(msg) {
  Timer.start(PERIODs);
  PowerDisable();
}

msgHandler rcvFail(msg) {
  Timer.stop();
  HOLDER = true;
  OUTf = OUTf ∪ {sender(msg)};
```

Figure 6. The Message Handler procedures.

```
eventHandler SensorEvent {
  PowerEnable();
  Timer.stop();
  HOLDER = true;
}

eventHandler Timer.fired() {
  if (power == Enable) {
    Timer.start(PERIODs);
    PowerDisable();
  } else {
    Timer.start(PERIODw);
    PowerEnable();
}
```

Figure 7. The Event Handler procedures.

In addition, for SWP we use a decreasing clock timer `CLOCK` that can be explicitly activated, deactivated and set to a given value, and two constant variables `PERIODw`, `PERIODs` provided by the implementer that indicate the length of the awake and sleeping periods of the particle. For example, in TinyOS, to create a timer that expires every `PERIODw` ms we use the statement `Timer.start(TIMER_REPEAT, PERIODw);` and `Timer.stop();` terminates the timer. Each time the timer expires, a `Timer.fired()` event is triggered that invokes a function implemented by the user.

Initially, the Boolean variables `HOLDER` and `EXECUTING` are set to `false`, the variable `PREVIOUS` is set to itself and the set `OUTf` is empty. For SWP, each particle is at *awake mode* with its `CLOCK` set to a period chosen randomly in the range (0, `PERIODw` + `PERIODs`]. In TinyOS this can be implemented using the method `Random.rand()` of the built-in 16-bit Linear Feedback Shift Register pseudo-random number generator.

The protocols use five types of messages: *info*($\mathcal{E}$), *fail*, *success*, *requestBeacon*, *beacon*. The first message type is used to propagate the actual information on the crucial event, while the next two (*fail* and *success*) are special control messages used to signify a failure or a success in the attempt to propagate *info*($\mathcal{E}$) towards the receiving wall $\mathcal{W}$. The *requestBeacon* and *beacon* messages are used by the search phase. Interestingly, in TinyOS, radio communication follows the *Active Message* (AM) *model*, in which each packet on the network specifies a *handler ID* that will be invoked on recipient nodes. When a message is received, the receive event associated with that handler ID is signaled. Thus we only need to define one message handler per message type. Figure 6 depicts the five message handlers implemented by the protocols.

Remark that the *beacon* message handler assumes that the communication module is capable of measuring the signal strength of the message received by executing the function `power(msg)`. Similarly, the function `sender(msg)` is used to extract the originator of a message, assuming that the message structure maintains such kind of information. Also, the function `remember(...)` adds the information to a temporary buffer. Using these primitives by sending a *requestBeacon* message the particle initiates the *search phase* and then, after waiting for a sufficient period of time (so that all neighbors respond to the request by sending a beacon), the `DetectNeighbors()` procedure processes the temporary buffer and returns a set containing those particles that responded to the broadcast of the search beacon, ordered by the distance of the particles (i.e., $d(p', p'')$).

Apart from the message event handlers, the protocols use two additional types of events: (i) `SensorEvent` created by the sensors of the particle when a crucial event is realized (i.e., when the particle is selected by the adversary $\mathcal{A}$) and (ii) `Timer.fired()` created by the `Timer` when the counting has finished. Figure 7 depicts the two generic event handlers implemented by our protocols. Remark that the `PowerDisable()` and `PowerEnable()` will force the particle to enter a "snooze" mode where only the *Timer* is active.

Based on the above event driven functionality, particle p executes continuously the **Main task**, shown in figure 8. TinyOS provides a two-level scheduling hierarchy consisting of *tasks* and *hardware event handlers*. Tasks are used to perform longer processing operations, such as background data processing, and can be preempted by hardware event handler. Remark that the `post` operation places the task on an internal task queue which is processed in FIFO order.

```
task main {
  if (HOLDER == true) {
      next = SenseNeighbours();
          if (next == nil)
            BackTrack();
          else {
            send(next)[info(E)];
            send(PREVIOUS)[success];
            HOLDER = false;
          }
  }

  post main();
}

procedure SenseNeighbors {
  send()[reqBeacon];

  Set tempSet = DetectNeighbors();
  if (tempSet == empty)
    return nil;

  Set out = tempSet − OUTf;
  if (out == empty)
    return nil;

  return out.first();
}

procedure Backtrack {
  send(PREVIOUS)[fail];
  HOLDER = false;
  Timer.start(PERIODs);
    PowerDisable();
}
```

Figure 8. The Main task and the SenseNeighbors, Backtrack procedures.

10. Experimental evaluation

In this section we report on four sets of experiments that aim to validate the theoretical analysis of the previous sections. We have implemented the three protocols using C++ and the data types for two-dimensional geometry of LEDA [22]. Each class is installed in an environment that generates sensor fields given some parameters (such as the area of the field, the distribution function used to drop the particles), and performs a network simulation for a given number of repetitions, a fixed number of particles and certain protocol parameters. After the execution of the simulation, the environment stores the results on files so that the measurements can be represented in a graphical way. Each experiment was conducted for more than 10,000 times in order to achieve good average results.

In the first set of experiments, we investigate (a) the impact of the angle α and (b) the number of targets found during the *search phase*, on the hops efficiency of the Local target protocol when considering the ideal case where the search phase always finds a particle (of sufficiently high battery) in $(-\alpha, \alpha)$ (we call the measured efficiency, the *ideal hops efficiency*). In figure 9 we observe that for both protocols, as $\alpha \to 0$, the ideal hops efficiency $C_h \to 1$. Actually, the ideal C_h initially decreases very fast with increasing α, while having a limiting behavior of no further significant improvement when $\alpha \leqslant 40$. Figure 10 shows the effect of finding more than one target during the search phase; as the number of targets increases, the ideal hops efficiency $C_h \to 1$. We note a similar threshold behavior, for a total number of 4 targets.

In the second set of experiments we study the performance of the LTP and m2TP protocols in more realistic cases by generating a variety of sensor fields in a 100 m $\times$ 100 m square.

In these fields, we drop $n \in [100, 5000]$ particles uniformly distributed on the smart-dust plane, i.e., $0.01 \leqslant d \leqslant 0.5$. Each smart dust particle has a radio range of $\mathcal{R} = 5$ m. For carrying out identical repetitions on our experiments we explicitly place a particle at position $(x, y) = (0, 50)$ and we assume that this particle detects the event. The wall is located at $x = 100$. In this set of experiments, the particle p' discovered in the search phase can be located anywhere within the cyclic sector defined by circles of radiuses $0, \mathcal{R}$ and angles $(-\alpha, \alpha)$. Note that this experimental setup is based on that used in [14,17,21]. Also, remark that the efficiency is *measured over the successful tries*, i.e., we do not take into account those runs that *backtracked*, however we keep track of the total number of times that the protocol was required to backtrack.

In figure 11 we observe that opposed to the ideal case (i.e., when the search phase always returns a particle on the semi-circle), we do not get significant improvement in the hops efficiency as the angle α is reduced. This is basically because the discovered particle p' might be close to p and thus the local improvement made is of limited significance. Note that the min-two uniform targets protocol (m2TP) achieves better efficiency compared to the local target protocol (LTP).

Figure 12 depicts the effect of density d on the hops efficiency of the two protocols. Interestingly, we observe that even for quite low density of particles (i.e., $d \leqslant 0.2$) the hops efficiency remains unaffected. This is a result of our choice not to include the failed searches in our measurements, that is, the measurements include only the search phases that resulted in finding a particle p closer to $\mathcal{W}$. To get a more complete view on the effect of density, figure 13 shows the failure rate (i.e., the number of times that the protocols backtracked) for different values of d. We observe that for low density (i.e., $d \leqslant 0.1$) both protocols almost always use the backtrack

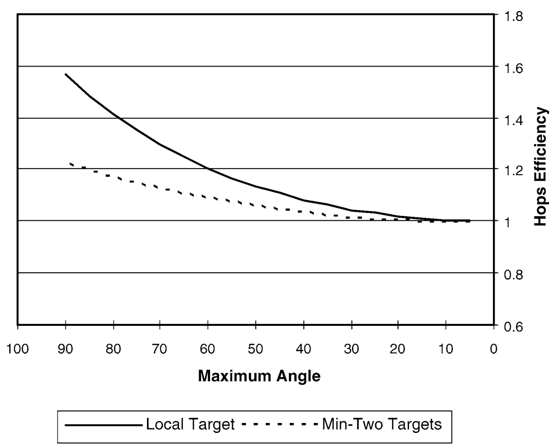

Figure 9. Ideal hops efficiency for angles $\alpha \in [5, 90]$.

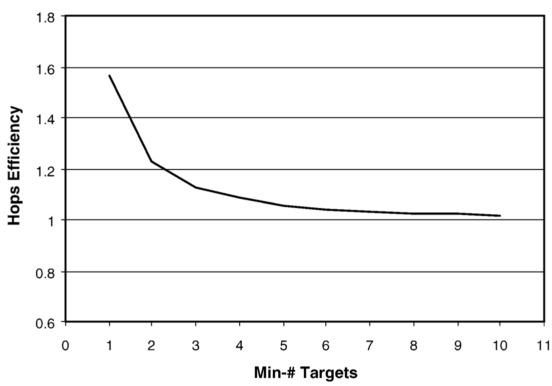

Figure 10. Ideal hops efficiency for different number of targets.

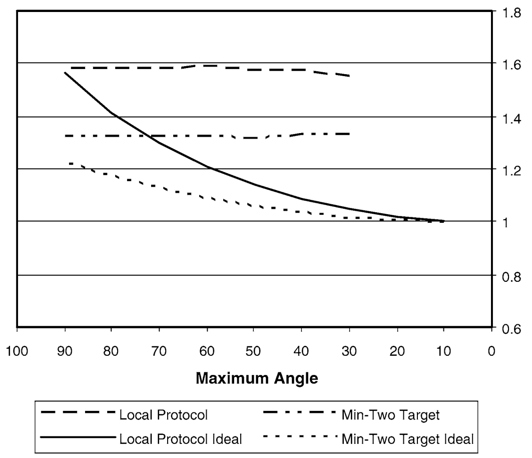

Figure 11. Hops efficiency for angles $\alpha \in [5, 90]$ for $d = 0.3$.

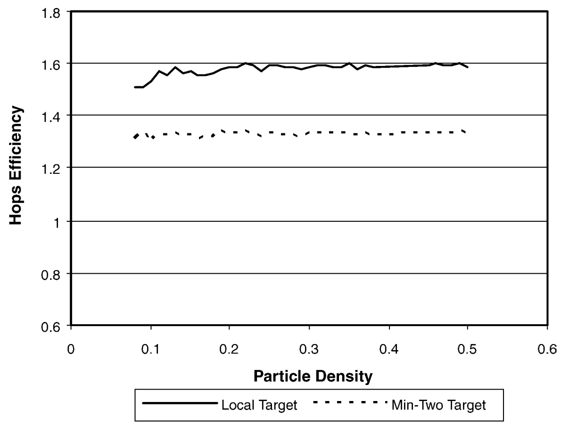

Figure 12. Hops efficiency for density $d \in [0.01, 0.5]$ and $\alpha = 90$.

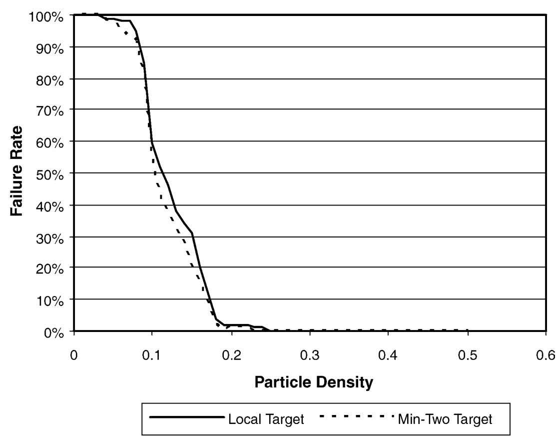

Figure 13. Failure rate for density $d \in [0.01, 0.5]$ and $\alpha = 90$.

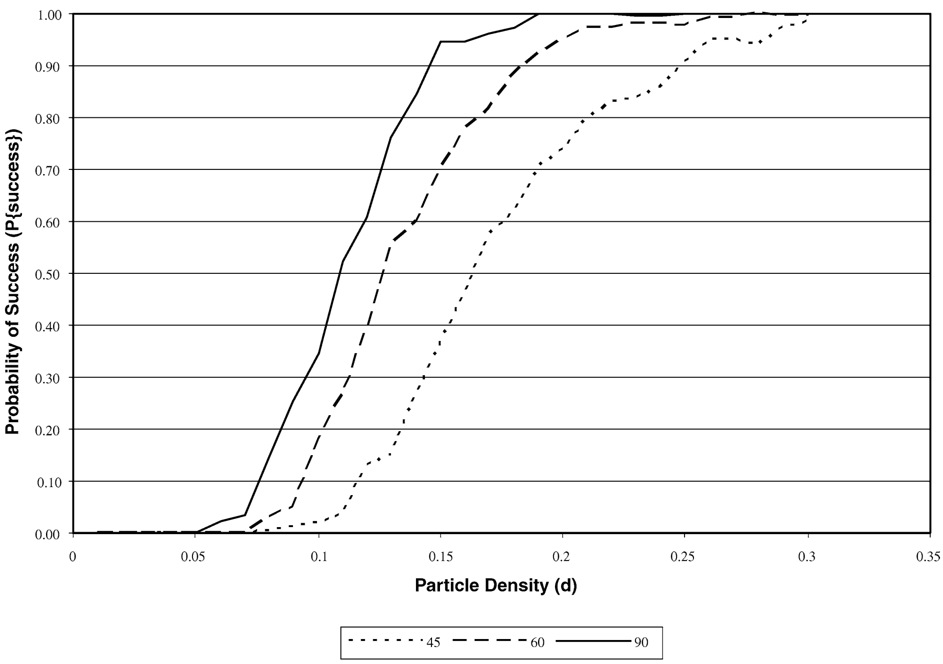

Figure 14. Probability of success ($\mathbb{P}\{success\}$) over particle density $d = [0.01, 0.3]$ for various angles $\alpha = \{45, 60, 90\}$, and random distribution.

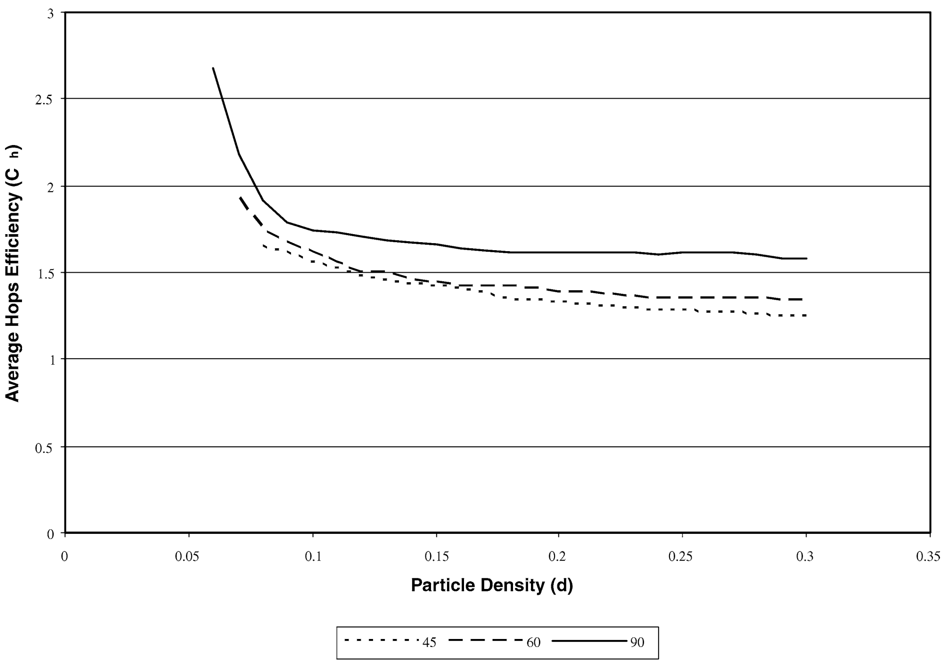

Figure 15. Average hops efficiency (C_h) over particle density $d = [0.01, 0.3]$ for various angles $\alpha = \{45, 60, 90\}$, and random distribution.

mechanism, while when $d \geqslant 0.2$ the failure rate drops very fast to zero. This can be justified by taking into account the average degree of each particle for various density d.

In the third set of experiments, we evaluate the performance of the SWP protocol in the case when all the particles remain awake (i.e., $en = 0$). We consider this a first step to investigate (a) the impact of the angle α and (b) the effect of the particles density d on the probability of success ($\mathbb{P}\{success\}$), hops efficiency (C_h) and average number of backtracks. The particle density was $0.01 \leqslant d \leqslant 0.3$ and used three different angles, $\alpha = \{45, 60, 90\}$ (in degrees). The reported experiments for the three different performance measures we considered are illustrated in figures 14–16.

Examining figure 14, that shows the probability of success ($\mathbb{P}\{success\}$), we first observe that for particle density $d < 0.05$ (i.e., throwing a small number of particles) the protocol fails to propagate the critical event (i.e., the success probability is zero). However, the probability of success *increases very fast* exhibiting a threshold-like behaviour as the particle density increases, and the protocol almost al-

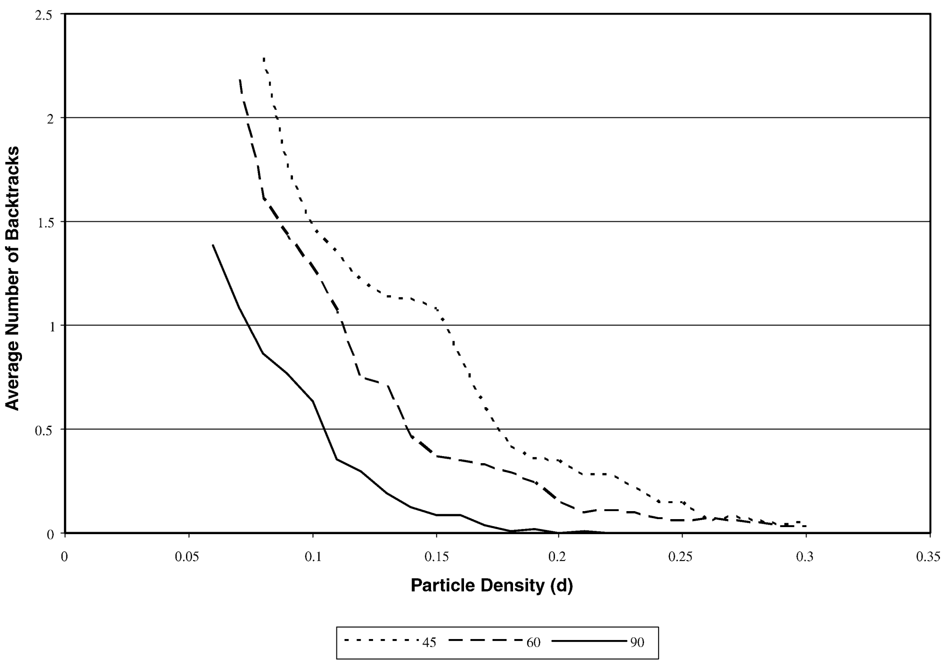

Figure 16. Average number of backtracks over particle density $d = [0.01, 0.3]$ for various angles $\alpha = \{45, 60, 90\}$, and random distribution.

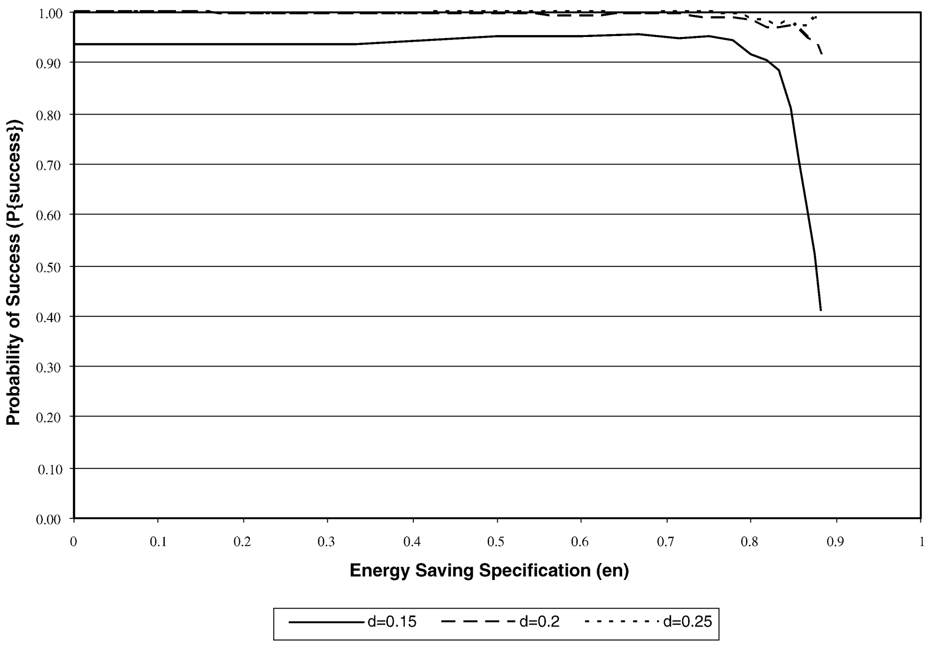

Figure 17. Probability of success ($\mathbb{P}\{success\}$) over en (where $en = s/(s+w)$) for various particle densities $d = \{0.15, 0.2, 0.25\}$, fixed angle $\alpha = 90$ and random distribution.

ways succeeds to propagate the critical event when $d > 0.15$. As expected (due to equation (3)), setting a smaller angle α reduces the probability of success in each density case since the number of particles that respond to the search phase gets smaller. So for $\alpha = 60$ the $\mathbb{P}\{success\}$ gets close to 1 when $d > 0.2$ while for $\alpha = 45$, $\mathbb{P}\{success\} \to 1$ when $d > 0.25$.

Regarding the average hops efficiency (C_h) we interestingly observe in figure 15 that even for a small particle density, the hops efficiency is close to the optimal. Actually, as the particle density crosses $d = 0.05$ (i.e., when $\mathbb{P}\{success\} > 0$) the hops efficiency gets close to 2.6. In fact, when $d = 0.1$, $C_h = 1.74$ while *no further gain is achieved* if we throw more particles (i.e., increase d). This is because of a sufficiently large density leads to many particles found in the search, of which particles already some are close to the vertical line. Similar results hold for $\alpha = 60$ and $\alpha = 45$, although at high particle densities, the hops efficiency is slightly better.

Finally, in figure 16 we can see the average number of backtracks performed by the protocol in the attempt to prop-

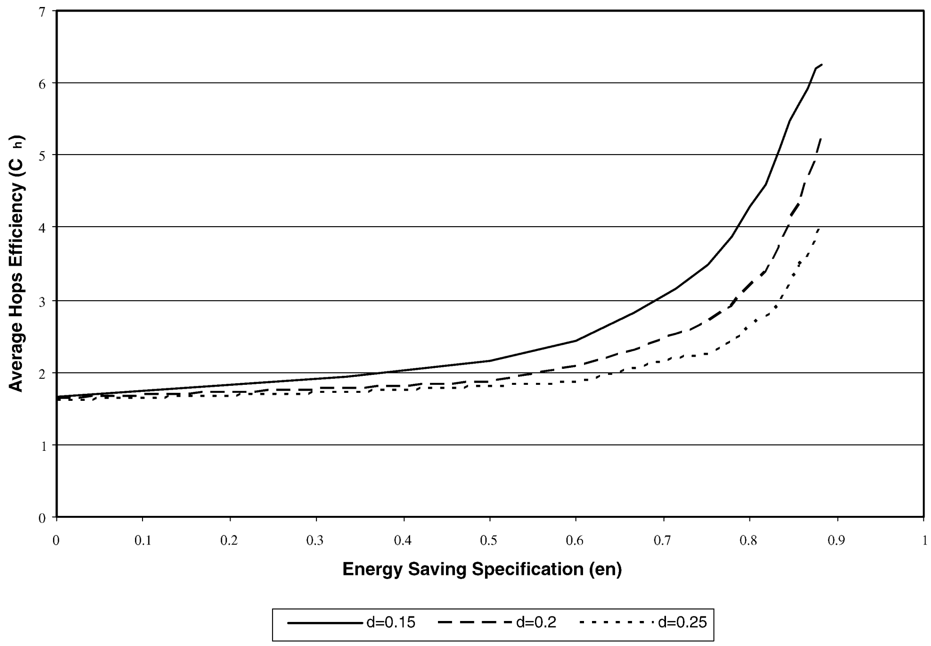

Figure 18. Average hops efficiency (C_h) over *en* (where $en = s/(s+w)$) for various particle densities $d = \{0.15, 0.2, 0.25\}$, fixed angle $\alpha = 90$ and random distribution.

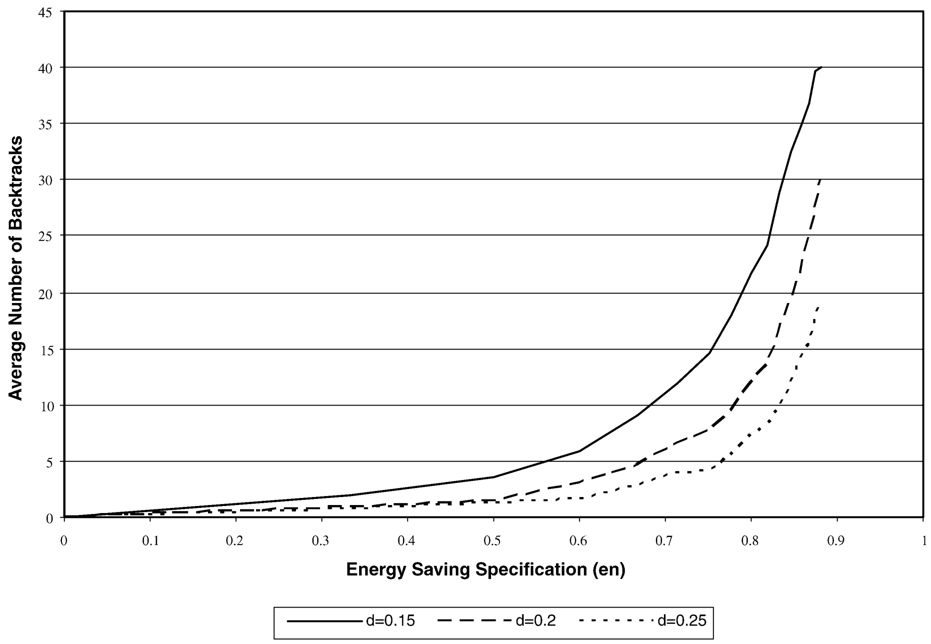

Figure 19. Average number of backtracks over *en* (where $en = s/(s+w)$) for various particle densities $d = \{0.15, 0.2, 0.25\}$, fixed angle $\alpha = 90$ and random distribution.

agate the critical event to the wall. We observe that for low density (i.e., $d \leqslant 0.1$) the protocols almost always uses the backtrack mechanism, while when $d \geqslant 1.5$ the number of backtracks performed drops very fast to zero. Furthermore, we observe that the number of backtracks is initially high but decreases with a fast rate as the particle density increases. This can be justified by taking into account the average degree of each particle for various density d. More specifically, when $\alpha = 90$ and $d = 0.15$ the protocol almost always succeeds in propagating the crucial event without the need to backtrack.

The last set of experiments aims to evaluate the impact of the energy saving specification *en* on the performance of the protocol. Again, we measure the probability of success ($\mathbb{P}\{success\}$), hops efficiency (C_h) and average number of backtracks over energy saving specification (en), for three different particle densities ($d = 0.15, 0.2, 0.25$) and three different angles $\alpha = \{45, 60, 90\}$ (in degrees). We have set the awake period $w = 2$ and the sleeping period $s \in [0, 15]$ thus making $en \in [0, 0.88]$ (recall that $en = s/(s+w)$). Figures 17–19 show the measured performance for the dif-

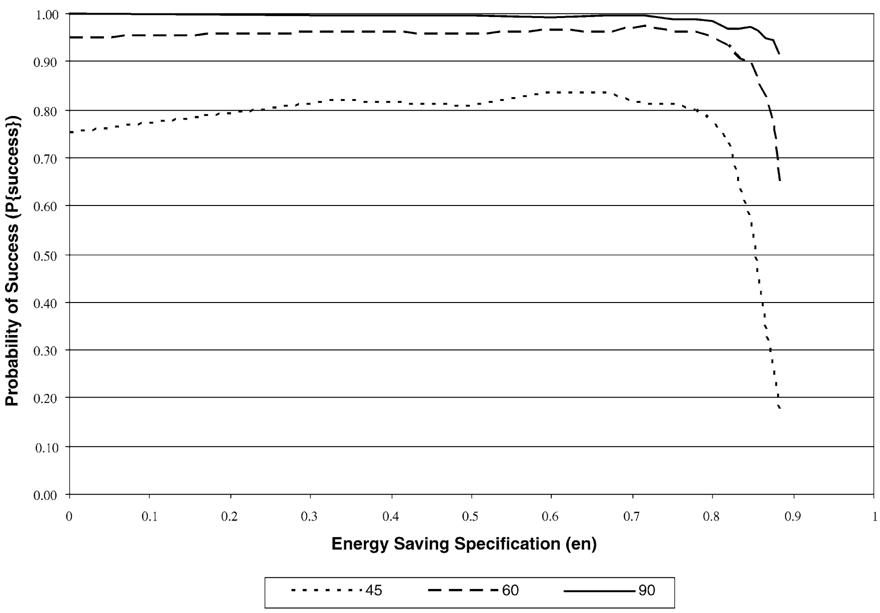

Figure 20. Probability of success ($\mathbb{P}\{success\}$) over en (where $en = s/(s + w)$) for various angles $\alpha = \{45, 60, 90\}$, fixed particle density $d = 0.2$ and random distribution.

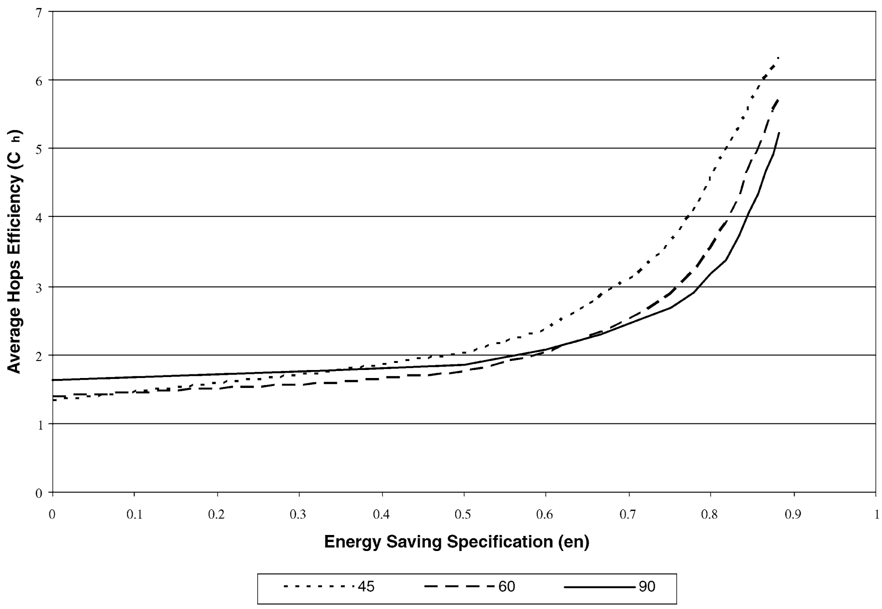

Figure 21. Average hops efficiency (C_h) over en (where $en = s/(s + w)$) for various angles $\alpha = \{45, 60, 90\}$, fixed particle density $d = 0.2$ and random distribution.

ferent particle densities d and figures 20–22 for the different angles α.

In figures 17 and 20, that show the probability of success for different particle densities d and angles α, we observe that the protocol experiences a threshold behavior when $en = 0.75$: when $en \leqslant 0.75$ the probability of success is close to 1 while for $en > 0.75$, $\mathbb{P}\{success\}$ drops very fast to zero. In other words, even if we set the particles to be awake only the 25% of each sleep–awake cycle, it does not affect the success of the protocol to propagate the information to $\mathcal{W}$. However, in figures 18 and 20 we observe a similar threshold behavior for the average hops efficiency when $en = 0.5$: the hops efficiency remains unaffected when $en \leqslant 0.5$ while it decreases very fast (i.e., C_h increases) when $en > 0.5$. Interestingly, figures 19 and 21 show that for the same threshold value ($en = 0.5$) the protocol almost always succeeds without the need to backtrack, while for $en > 0.5$ the number of backtracks increases very fast with en. Thus, although for $en \leqslant 0.8$ the probability of success is close to 1, setting the energy saving specification to $en = 0.5$ seems to be more reasonable. This leads to the conclusion that by setting the particles to be active only the 50% of the overall period for which the protocol is executed, we manage to decrease the energy requirements while

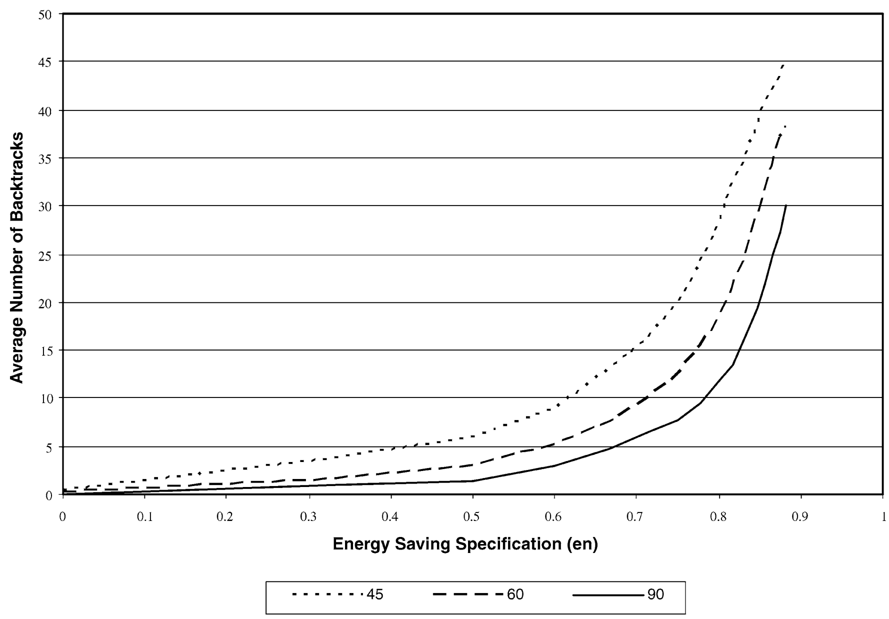

Figure 22. Average number of backtracks over *en* (where $en = s/(s+w)$) for various angles $\alpha = \{45, 60, 90\}$, fixed particle density $d = 0.2$ and random distribution.

keeping the hops efficiency (and thus time efficiency) unaffected.

11. Conclusions and future work

We presented here a model for Smart Dust and three basic protocols (and their average case performance) for local detection and propagation. We plan to investigate protocols that trade-off hops efficiency and time, as well as study the fault-tolerance of protocols as a function of smart dust parameters (such as density of the cloud, the energy saving characteristics, etc.). We also intend to investigate alternative backtrack mechanisms and study their effect on the efficiency and fault-tolerance of the protocol. Also, we are currently working towards the design of local protocols than can monitor the spreading of a time-sequence of events (i.e., tracking protocols). We plan to provide performance comparisons with other protocols mentioned in the related work section. Finally, we plan to also explicitly introduce sensor faults and study the performance (efficiency, fault-tolerance) of our protocols in this case.

References

[1] T. Antoniou, A. Boukerche, I. Chatzigiannakis, G. Mylonas and S. Nikoletseas, A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range, in: *Proc. 37th ACM/IEEE Annual Simulation Symposium ANSS*, Arlington, VA, USA, April 18–22 (2004) pp. 167–177.

[2] Berkeley Wireless Research Center (2001), `http://bwrc.eecs.berkeley.edu`

[3] A. Boukerche and S. Nikoletseas, Protocols for data propagation in wireless sensor networks: a survey, in: *Wireless Communications Systems and Networks*, ed. M. Guizani (Kluwer Academic, 2004) chapter 2, pp. 23–51.

[4] I. Chatzigiannakis, T. Dimitriou, S. Nikoletseas and P. Spirakis, A probabilistic algorithm for efficient and robust data propagation in smart dust networks, in: *Proc. 5th European Wireless Conference (EW 2004)*, Barcelona, Spain, February 24–27 (2004) pp. 344–350.

[5] I. Chatzigiannakis, T. Dimitriou, M. Mavronicolas, S. Nikoletseas and P. Spirakis, A comparative study of protocols for efficient data propagation in smart dust networks, in: *Proc. International Conference on Parallel and Distributed Computing, EUPOPAR 2003* (2003). Also in Parallel Processing Letters 13(4) (2003) 615–627.

[6] I. Chatzigiannakis and S. Nikoletseas, A sleep–awake protocol for information propagation in smart dust networks, in: *Proc. 3nd International Workshop on Mobile and Ad-Hoc Networks*, *17th Annual International Parallel and Distributed Processing Symposium (IPDPS 2003)*, Nice, France, April 21–25 (2003) p. 225.

[7] I. Chatzigiannakis, S. Nikoletseas and P. Spirakis, An efficient communication strategy for ad-hoc mobile networks, in: *Proc. 15th International Symposium on Distributed Computing (DISC 2001)*, pp. 285–299. See also brief announcement in: *Proc. 20th Annual Symposium on Principles of Distributed Computing (PODC 2001)*, pp. 320–322.

[8] I. Chatzigiannakis, S. Nikoletseas and P. Spirakis, On the average and worst-case efficiency of some new distributed communication and control algorithms for ad-hoc mobile networks, in: *Proc. 1st ACM Workshop on Principles of Mobile Computing (POMC 2001)*, pp. 1–19, invited paper.

[9] I. Chatzigiannakis, S. Nikoletseas and P. Spirakis, Smart dust protocols for local detection and propagation, in: *Proc. 2nd ACM Workshop on Principles of Mobile Computing (POMC 2002)*, pp. 9–16.

[10] I. Chatzigiannakis, A. Kinalis and S. Nikoletseas, Wireless sensor networks protocols for efficient collision avoidance in multi-path data propagation, in: *Proc. of the ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN* (2004).

[11] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, Next century challenges: scalable coordination in sensor networks, in: *Proc. 5th ACM/IEEE International Conference on Mobile Computing (MOBICOM 1999)* (1999) pp. 263–270.

[12] H. Euthimiou, S. Nikoletseas and J. Rolim, Energy balanced data propagation in wireless sensor networks, in: *Proc. 4th International Workshop on Algorithms for Wireless, Mobile, Ad-Hoc and Sensor Networks (IPDPS/WMAN'04)* (Computer Society Press, 2004) p. 225.

[13] W. Feller, *An Introduction to Probability Theory and Its Applications* (Wiley, New York, 1957).

[14] W.R. Heinzelman, A. Chandrakasan and H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: *Proc. 33rd Hawaii International Conference on System Sciences (HICSS 2000)* (2000) p. 8020.
[15] W.R. Heinzelman, J. Kulik and H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, in: *Proc. 5th ACM/IEEE International Conference on Mobile Computing (MOBICOM 1999)* (1999) pp. 174–185.
[16] S.E.A. Hollar, COTS dust, M.Sc. Thesis in engineering–mechanical engineering, University of California, Berkeley, USA (2000).
[17] C. Intanagonwiwat, D. Estrin, R. Govindan and J. Heidemann, Impact of network density on data aggregation in wireless sensor networks, Technical Report 01-750, University of Southern California Computer Science Department (November, 2001).
[18] C. Intanagonwiwat, R. Govindan and D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in: *Proc. 6th ACM/IEEE International Conference on Mobile Computing (MOBICOM 2000)* (2000) pp. 56–67.
[19] B. Karp, Geographic routing for wireless networks, Ph.D. Dissertation, Harvard University, Cambridge, MA (2000).
[20] μ-adaptive multi-domain power aware sensors (April 2001), `http://www-mtl.mit.edu/research/icsystems/uamps`
[21] A. Manjeshwar and D.P. Agrawal, TEEN: A routing protocol for enhanced efficiency in wireless sensor networks, in: *Proc. 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing (PDCIWNMC'02), 16th Annual International Parallel and Distributed Processing Symposium (IPDPS 2002)*, Fort Lauderdale, FL, USA, April 15–19, 2002 (2004) p. 1956.
[22] K. Mehlhorn and S. Näher, *LEDA: A Platform for Combinatorial and Geometric Computing* (Cambridge University Press, 1999).
[23] S. Nikoletseas, I. Chatzigiannakis, H. Euthimiou, A. Kinalis, A. Antoniou and G. Mylonas, Energy efficient protocols for sensing multiple events in smart dust networks, in: *Proc. 37th ACM/IEEE Annual Simulation Symposium (ANSS)* (2004).
[24] C.E. Perkins, *Ad Hoc Networking* (Addison-Wesley, Boston, MA, 2001). (Reference to book spans across different chapters since the text is discussing the case of porting existing mobile ad-hoc routing protocols to smart dust networks.)
[25] S.M. Ross, *Stochastic Processes*, 2nd ed. (Wiley, 1995).
[26] C. Schurgers, V. Tsiatsis and M.B. Srivastava, STEM: Topology management for energy efficient sensor networks, in: *Proc. IEEE Aerospace Conference 2002*, Vol. 3 (2002) pp. 1099–1108.
[27] S. Singh and C.S. Raghavendra, PAMAS: Power aware multi-access protocol with signalling for ad hoc networks, ACM Computer Communication Review 28(3) (1998) 5–25.
[28] TinyOS, A component-based os for the network sensor regime (October, 2002), `http://webs.cs.berkeley.edu/tos/`
[29] Wireless Integrated Sensor Networks (April 2001), `http://www.janet.ucla.edu/WINS/`
[30] W. Ye, J. Heidemann and D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: *Proc. 12th IEEE International Conference on Computer Networks (INFOCOM 2002)* (2002) pp. 947–957.

Ioannis Chatzigiannakis is a Researcher of Research Unit 1 ("Foundations of Computer Science, Relevant Technologies and Applications") at the Computer Technology Institute (CTI), Greece. He has received his B.Eng. degree from the University of Kent, UK in 1997 and his Ph.D. degree from the Computer Engineering and Informatics Department of Patras University, Greece in 2003, under the supervision of Prof. Paul Spirakis. His research interests include distributed computing, mobile computing and algorithmic engineering. He has served as an external reviewer in major international conferences. He has participated in several European Union funded R&D projects, and worked in the private sector.
E-mail: ichatz@cti.gr

Sotiris E. Nikoletseas is currently a Lecturer Professor at the Computer Engineering and Informatics Department of Patras University, Greece and also a Senior Researcher and Director of Research Unit 1 ("Foundations of Computer Science, Relevant Technologies and Applications") at the Computer Technology Institute (CTI), Greece. His research interests include probabilistic techniques and random graphs, average case analysis of graph algorithms and randomized algorithms, algorithmic applications of probabilistic techniques in distributed computing (focus on ad-hoc mobile networks and wireless sensor networks), algorithmic applications of combinatorial and probabilistic techniques in fundamental aspects of modern networks (focus on network reliability and stability), approximation algorithms for computationally hard problems. He has published over 80 scientific articles in major international conferences and journals and has co-authored a book on probabilistic techniques, a chapter in the *Handbook of Randomized Computing* (Kluwer Academic) and several chapters in books of international circulation in topics related to distributed computing. He has been invited speaker in international scientific events and Universities and he has delivered several tutorials and keynote talks. He has been a reviewer for important computer science journals and has served in the Program and Organizing Committees of International Conferences and Workshops. He has participated in many European Union funded R&D projects.
E-mail: nikole@cti.gr

Paul G. Spirakis born in 1955, got his Ph.D. from Harvard University in 1982. He became an Assistant Professor at NYU (the Courant Institute) the same year. He was then elected as a tenured Associate Professor at Patras University, Greece, in the Department of Computer Engineering and Informatics. He became a Full Professor in the same Department in 1990. He served as the Chairman of the Department of Computer Engineering and Informatics for six years. Since 1996 he is the Director of the Research and Academic Computer Technology Institute (RACTI) of Greece. His research interests include algorithms, probabilistic techniques, distributed and parallel computing and average case analysis. Recently P. Spirakis is active in the area of algorithmic aspects of game theory and also in algorithmic aspects of ad-hoc and sensor networks. Paul Spirakis has won the top prize of the Greek Mathematical Society in 1973 as a student. Since then he won several awards and many competitive grants. He was appointed a distinguished Visiting Researcher of Max Planck Informatik. He served as the National Representative of Research in Informatics in the EU for four years and is now a Member of the ISTAG group, a high level EU group responsible for Research Planning in Informatics. He is currently one of the two Vice Chairs of the European Association for Theoretical Computer Science (EATCS). He is a high level consultant of the Greek State and Industry in Informatics. Paul Spirakis has published extensively in many journals and conferences of his field (more than 150 publications currently). The journals and conferences where his work appears are among the most competitive worldwide. He coauthored two books with Cambridge University Press and seven books in Greek. Paul Spirakis serves as an Editor in many scientific journals of computer science and usually in the scientific commitees of some of the most prestigeous computer science conferences. He also serves periodically as a high level research evaluator for the EU and Greece.
E-mail: spirakis@cti.gr