A Sleep-Awake Protocol for Information Propagation in Smart Dust Networks *

Ioannis Chatzigiannakis and Sotiris Nikoletseas
1. Computer Technology Institute
P.O. Box 1122, 26221 Patras, Greece
{ichatz,nikole}@cti.gr
2. Computer Engineering and Informatics Department
University of Patras, 26500 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 continue (see [4]) our effort towards the research on smart dust from a basic algorithmic point of view. Under a simple but realistic model for smart dust we present an interesting problem, which is how to propagate efficiently information on an event detected locally. Then we present a new smart dust protocol, which we call the "Sleep-Awake" protocol, for information propagation that explicitly *uses the energy saving features (i.e. the alteration of sleeping and awake time periods) of the smart dust particles. By using both probabilistic some first analysis and extensive experiments, we provide some first concrete results for the success probability and the time and energy efficiency of the protocol, in terms of parameters of the smart dust network. We note that the study of the interplay of these parameters allows us to program the smart dust network characteristics accordingly.*

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 cooperate 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, [5] 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 [12], 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 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 [1–4], 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 both by analytic and experimental means, concrete results for the efficiency of a power-aware smart dust protocols using a probabilistic approach. We focus in this paper on the efficient use of smart dust in local detection and propagation protocols. We first present 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 measures (the expected "hops efficiency" and the probability of success) for the average case analysis of protocols for this problem.

Recent Related Work of Our Team: This work continues the line of research recently initiated by our team in [4]. For the local detection and propagation problem using smart dust, in [4] we have provided two protocols. Both 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 provided a rigorous average case analysis for the efficiency of these protocols. We considered 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. The first protocol (the "local target" protocol) uses a fast and cheap search phase which is assumed to always return a nearby (within broadcast range) particle closer to the authorities, uniformly in some range. It achieves a propagation time and an energy consumption whose expected ratio over the optimal solutions is at most $\pi/2 \simeq 1.57$. The second protocol (the "min-two uniform targets" protocol) 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 version of the first protocol, and has an expected time and energy ratio over the optimal solutions which is at most $\pi^2/8 \simeq 1.24$. We have also provided tight upper bounds to the distribution of the number of particle to particle data transmissions of a generalized target protocol.

Our Results: In this work, we propose a *new* protocol which we call the "Sleep-Awake" protocol. In contrast to the protocols we have presented in [4], this new protocol *explicitly* uses the energy saving characteristics, such as the alteration of sleep and awake time periods, of smart dust particles. The success (eventual information propagation to the wall) as well as the time and energy efficiency of this protocol, depend on a variety of parameters of the smart dust network, such as the density of grain particles and their distribution, the distribution of the sleeping and awake periods, the search angle α etc. (these parameters are precisely defined below). We here provide some first results on the interplay of such parameters. By using both first analytic and extensive experimental means, we in particular 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. We note that these results allow to program the smart dust network characteristics accordingly.

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. [16] 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 [19] 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 ([15]) 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. [11] 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. [7] 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. [13] introduces a new energy efficient routing protocol that does not provide periodic data monitoring (as in [7]), 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 [7] achieves less energy consumption and response time.

A family of negotiation-based information dissemination protocols suitable for wireless sensor networks is presented in [8]. 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 [9], 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. [10] 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 work is using a similar approach to the recent work of [17], where a new technique called Sparse Topology and Energy Management (STEM) is proposed that aggressively puts nodes to sleep. Interestingly, the analysis and experiments show improvements of nearly two orders of magnitude compared to sensor networks without topology management.

2 The Model

Smart dust is comprised of a vast number of ultra-small homogenous sensors, which we call "grain" particles. Each 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 fig. 1). Note that a two-dimensional setting is also used in [7–10, 13].

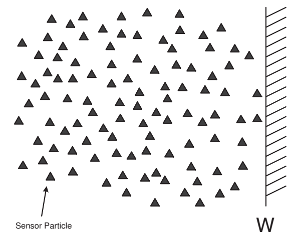

Figure 1. A Smart Dust Cloud

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 (mean and distribution) 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 [7, 13]). In contrast to [9, 11], 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 [6] a thorough comparative study and description of smart dust systems is given, from the technological point of view. In the Appendix we report some basic technical characteristics which we took into account when defining the model of smart dust we use here.

3 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 3.1 Let $h_{opt}(p,\, \mathcal{W})$ 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

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

Clearly, the number of hops (transmissions) needed characterizes the energy consumption and the time needed to propagate the information $\mathcal{E}$ to the wall. In fact, if we assume that each data transmission (one hop) takes constant time t, the "hops" and time efficiency of our protocols coincide in this case. Remark that $h_{opt} \;=\; \left\lceil \frac{d(p,\,\mathcal{W})}{\mathcal{R}} \right\rceil$, 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 does not necessarily 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.

4 The "Sleep-Awake" Protocol

We now present a new protocol for smart dust networks which we call the "sleep-awake" protocol. In contrast to the previous protocols (see [4]), 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.

Let $d(p_i,\, p_j)$ the distance (along the corresponding vertical lines towards $\mathcal{W}$) of particles $p_i,\, p_j$. 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 of angle α towards the wall in order to discover a particle nearer to $\mathcal{W}$ than itself. (i.e. a particle p'' where $d(p'',\, \mathcal{W}) \;<\; d(p',\, \mathcal{W})$). In case that more than one particles p'' are found, the best (w.r.t. vertical progress towards the wall) is selected.
- *Direct Transmission Phase*: Then, p' sends *info($\mathcal{E}$)* to p'' via a direct line (laser) time consuming transmission and sends a *success* message to p (i.e. to the particle that it originally received the information from).
- *Backtrack Phase*: If the *search phase* fails to discover a particle nearer to $\mathcal{W}$, then p' sends a *fail* message to p.

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.

Note that the search phase returns a list of particles that responded to the broadcast of the beacon and are positioned inside a cyclic sector defined by circles of radiuses $\mathcal{R} - \Delta\mathcal{R}$, $\mathcal{R}$ and arcs of angle $2a$ around the direct line from p' vertical to $\mathcal{W}$ on these circles. The angle α and the range $\Delta\mathcal{R}$ are provided by the implementer by taking into account the density of the smart cloud. Also note that the maximum distance $d(p',\, p'')$ is $\mathcal{R}$, i.e. the beacon transmission range (for a graphical representation see figures 2,3 where for symbol α we use a).

We now proceed by providing a more detailed description of the protocol implementation:

Message Structures: The protocol uses three types of messages: $info(\mathcal{E})$, $fail$ and $success$. The first message type is used to propagate the actual information on the crucial event, while the other 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 $\mathcal{W}$.

Data Structures at a Particle: At every particle, we use two boolean variables $HOLDER$ and $EXECUTING$ to denote the status of the particle. These variables are true only if the site holds *info($\mathcal{E}$)* (the information about the realization of the crucial event $\mathcal{E}$ to be propagated) or if the site has just transmitted *info($\mathcal{E}$)* to a neighboring particle, respectively. A variable

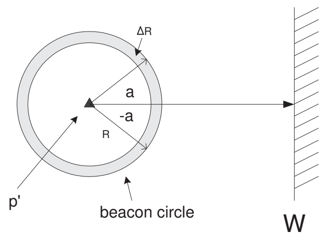

Figure 2. The Search Phase

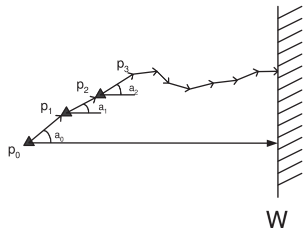

Figure 3. Example of a Transmission

$PREVIOUS$ records the particle from which *info($\mathcal{E}$)* was received. A decreasing clock timer $CLOCK$ that can be explicitly activated, deactivated and set to a given value. Two constant variables $PERIOD_w$, $PERIOD_s$ provided by the implementer that indicate the length of the awake and sleeping periods of the particle. Finally, a set $OUTF$ used to store any particle that failed to propagate a message towards the wall (this set is used for back-tracking purposes).

Initialization: Initially, each particle is at *awake mode* with its $CLOCK$ set to a period chosen randomly in the range $(0, PERIOD_w + PERIOD_s]$. This is done using a random bit generator, or it can be hardcoded into the particle by the manufacturer using a constant variable $PERIOD_i$. The boolean variables $HOLDER$ and $EXECUTING$ are set to `false`, the variable $PREVIOUS$ is set to itself and the set $OUTF$ is empty.

General Comments: We assume an event driven operating system suitable for smart dust (see [18]) that uses *event handlers* to process any event created by the subsystems of the particle. Our protocol uses three different types of events: (i) `Clock-Event` created by the $CLOCK$ when it reaches value 0, (ii) `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 (iii) `MessageEvent` created whenever a message has been received through the communication subsystem. Figure 4 depicts the five event handlers implemented by our protocol. Remark that when a `MessageEvent` is created, a generic event handler is invoked that, depending on the type of the message, makes a call to the **ReceiveInfoHandler**, **ReceiveFailHandler** and **ReceiveSuccessHandler**.

Based on the above event driven functionality, particle p executes continuously the **SleepAwake procedure**, shown in figure 5. Remark that the **SenseNeighbors** procedure uses the **DetectNeighbors** procedure that actually implements the *search phase* and that returns a set containing those particles that responded to the broadcast of the search beacon and that lie within the arc $(-\alpha, \alpha)$. This set is ordered by the distance of the particles (i.e. $d(p', p'')$) and the operation `.first()` will return the particle with the larger distance (i.e. located closer to $\mathcal{W}$). Finally, the pseudo-code provided uses function **sender**(`Message`) to extract the originator of a message, assuming that the message structure maintains such kind of information.

procedure SensorEventHandler
begin
  $pause\ CLOCK$;
  $HOLDER = true$;
end

procedure ClockEventHandler
begin
  if (*mode* == *awake*)
    begin
      $CLOCK = PERIOD_s$;
      $mode = sleeping$;
    end
  else
    begin
      $CLOCK = PERIOD_w$;
      $mode = awake$;
    end
end
procedure ReceiveInfoHandler
begin
  $pause\ CLOCK$;
  $HOLDER = true$;
  $PREVIOUS = sender(info(\mathcal{E}))$;
end

procedure ReceiveSuccessHandler
begin
  $EXECUTING = false$;
  $resume\ CLOCK$;
end

procedure ReceiveFailHandler
begin
  $pause\ CLOCK$;
  $HOLDER = true$;
  $EXECUTING = false$;
  $OUTF = OUTF \cup \{sender(fail)\}$;
end

Figure 4. The Event Handler procedures

4.1 Analysis

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 per m^2 and their distribution, the distribution of sleeping and awake time periods and, of course, on the angle α of the search beacon.

```
procedure SleepAwake
begin
  while (true) do
    if (HOLDER == true)
      begin
        Particle next = SenseNeighbours();
        if (next == nil)
          begin
            BackTrack();
          end
        else
          begin
            transmit info(ℰ) to next;
            transmit success to PREVIOUS;
            EXECUTING = true;
          end
      end
  loop
end

procedure SenseNeighbors
begin
  Set tempSet = DetectNeighbors();
  if (tempSet == empty)
    begin
      return nil;
    end
  Set out = tempSet − OUTF;
  if (out == empty)
    begin
      return nil;
    end
  return OUT.first();
end

procedure Backtrack
begin
  transmit fail to PREVIOUS;
  HOLDER = false;
  resume CLOCK;
end
```

Figure 5. The SleepAwake, SenseNeighbors and Backtrack procedures

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

$$\begin{aligned} N &= d \cdot \left(\frac{\alpha}{\pi} \pi \mathcal{R}^2 - \frac{\alpha}{\pi} \pi (\mathcal{R} - \Delta\mathcal{R})^2 \right) \\ &\simeq \alpha(2\mathcal{R}\Delta\mathcal{R} - \Delta\mathcal{R}^2) d \simeq 2\alpha\mathcal{R}\Delta\mathcal{R} d \end{aligned} \tag{1}$$

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 hard-coded 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:

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

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

Definition 4.1 *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 $P_1 = 1 - e^{-n}$. Then, by the Bernoulli inequality, we have $P_1^{E(h)} \geq 1 - E(h)e^{-n}$. 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.

5 Experimental Evaluation

We evaluate the performance of the "sleep-awake" protocol by conducting two set of experiments. Our implementation follows closely the pseudo-code provided in figures 5 and 4. The protocol has been implemented as C++ classes using the data types for two-dimensional geometry of LEDA [14]. 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.

In our experiments, we generate a variety of sensor fields in a 100m by 100m square. In these fields, we drop $n \in [100, 3000]$ particles uniformly distributed on the smart-dust plane, i.e. $0.01 \leq d \leq 0.3$. Each smart dust particle has a fixed radio range of

$\mathcal{R} = 5m$. 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 $x = 100$. Note that this experimental setup is based on and extends that used in [7, 10, 13]. We repeated each experiment for more than 5,000 times in order to achieve good average results.

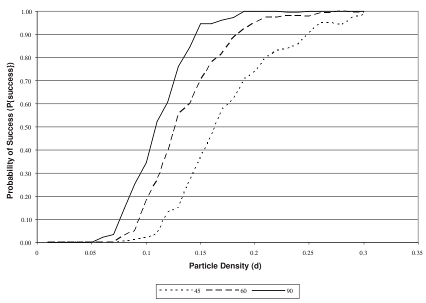

Figure 6. Probability of Success ($\mathbb{P}\{success\}$) over particle density $d = [0.01, 0.3]$ for various angles $\alpha = \{45, 60, 90\}$, and random distribution.

In the first set of experiments, we evaluate the performance of the 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 $0.01 \leq d \leq 0.3$ 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 6–8.

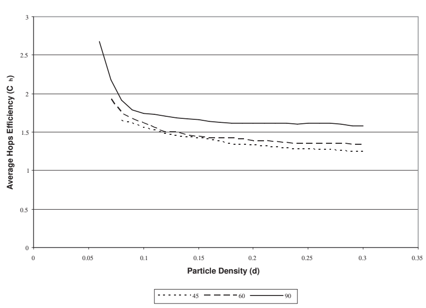

Figure 7. Average Hops Efficiency (C_h) over particle density $d = [0.01, 0.3]$ for various angles $\alpha = \{45, 60, 90\}$, and random distribution.

By examining Figure 6, 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 always succeeds to propagate the critical event when $d > 0.15$. As expected (due to Eq. 1), 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 $d > 0.25$.

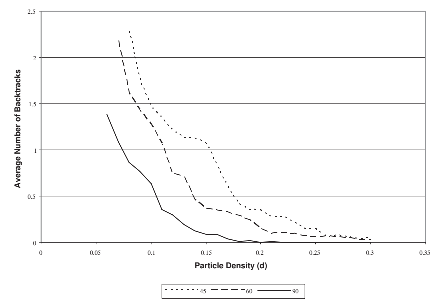

Figure 8. Average Number of Backtracks over particle density $d = [0.01, 0.3]$ for various angles $\alpha = \{45, 60, 90\}$, and random distribution.

Regarding the average hops efficiency (C_h) we interestingly observe in Figure 7 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 8 we can see the average number of backtracks performed by the protocol in the attempt to propagate the critical event to the wall. We observe that for low density (i.e. $d \leq 0.1$) the protocols almost always uses the backtrack mechanism, while when $d \geq 1.5$ 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 second 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 $en = \frac{s}{s+w}$). Figures 9–11 show the measured performance for

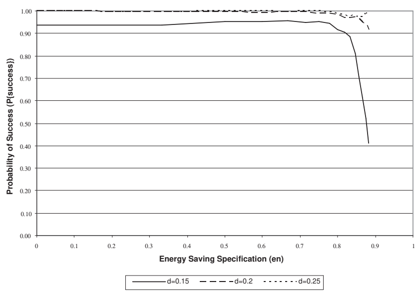

Figure 9. Probability of Success ($\mathbb{P}\{success\}$) over en (where $en = \frac{s}{s+w}$) for various particle densities $d = \{0.15, 0.2, 0.25\}$, fixed angle $\alpha = 90$ and random distribution.

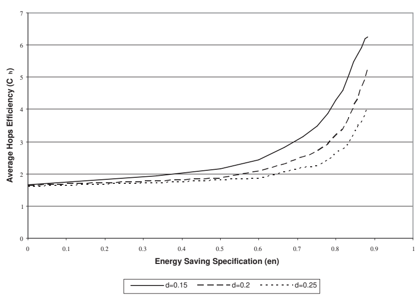

Figure 10. Average Hops Efficiency (C_h) over en (where $en = \frac{s}{s+w}$) for various particle densities $d = \{0.15, 0.2, 0.25\}$, fixed angle $\alpha = 90$ and random distribution.

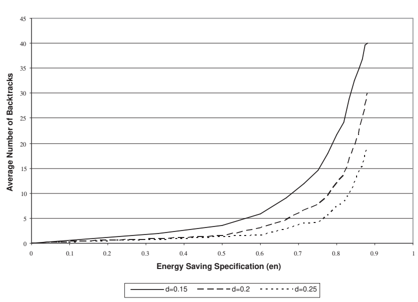

Figure 11. Average Number of Backtracks over en (where $en = \frac{s}{s+w}$) for various particle densities $d = \{0.15, 0.2, 0.25\}$, fixed angle $\alpha = 90$ and random distribution.

the different particle densities d and Figures 12–14 for the different angles α.

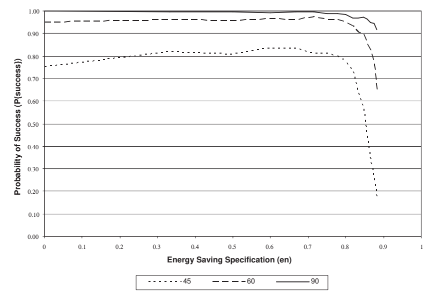

Figure 12. Probability of Success ($\mathbb{P}\{success\}$) over en (where $en = \frac{s}{s+w}$) for various angles $\alpha = \{45, 60, 90\}$, fixed particle density $d = 0.2$ and random distribution.

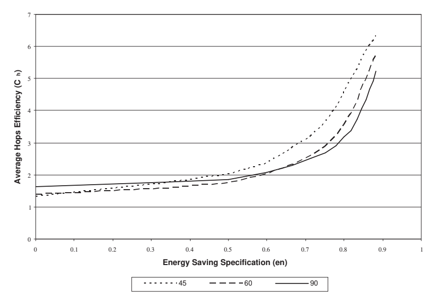

Figure 13. Average Hops Efficiency (C_h) over en (where $en = \frac{s}{s+w}$) for various angles $\alpha = \{45, 60, 90\}$, fixed particle density $d = 0.2$ and random distribution.

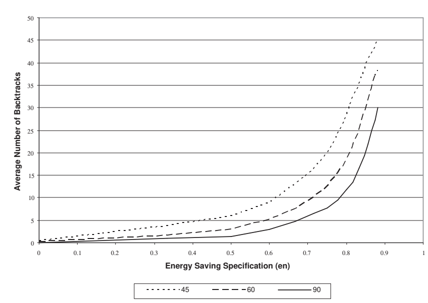

Figure 14. Average Number of Backtracks over en (where $en = \frac{s}{s+w}$) for various angles $\alpha = \{45, 60, 90\}$, fixed particle density $d = 0.2$ and random distribution.

In Figures 9 and 12, 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

$en \leq 0.75$ 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 10 and 12 we observe a similar threshold behavior for the average hops efficiency when $en = 0.5$: the hops efficiency remains unaffected when $en \leq 0.5$ while it decreases very fast (i.e. C_h increases) when $en > 0.5$. Interestingly, Figures 11 and 13 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 $en \leq 0.8$ 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 keeping the hops efficiency (and thus time efficiency) unaffected.

6 Conclusions and Future Work

We presented here a new protocol for information propagation in Smart Dust networks. Analysis and experimental results indicate that the new protocol reduces the overall energy requirements while maintaining high hops and time efficiency. We plan 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). Finally, we plan to provide performance comparisons with other protocols mentioned in the related work section.

References

[1] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis: Distributed Communication Algorithms for Ad-hoc Mobile Networks. In the Journal of Parallel and Distributed Computing (JPDC), Special Issue on Mobile Ad-hoc Networking and Computing (to appear in 2002). Also in *Proc. 15th International Symposium on Distributed Computing* – DISC'2001. Lecture Notes in Computer Science, Volume 2180 (Springer-Verlag, 2001), pp. 285–299.

[2] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis: An Efficient Routing Protocol for Hierarchical Ad-Hoc Mobile Networks. In *Proc. 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, satellite workshop of *15th Annual International Parallel & Distributed Processing Symposium* – IPDPS'01.

[3] I. Chatzigiannakis and S. Nikoletseas: An Adaptive Compulsory Protocol for Basic Communication in Highly Changing Ad-hoc Mobile Networks. In *Proc. 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, satellite workshop of *16th Annual International Parallel & Distributed Processing Symposium* – IPDPS'02.

[4] I. Chatzigiannakis, S. Nikoletseas and P. Spirakis: Smart Dust Protocols for Local Detection and Propagation. To Appear in *Proc. 2nd ACM Workshop on Principles of Mobile Computing* – POMC'2002.

[5] 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.

[6] S.E.A. Hollar: COTS Dust. Msc. Thesis in Engineering-Mechanical Engineering, University of California, Berkeley, USA, 2000.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] B. Karp: Geographic Routing for Wireless Networks. Ph.D. Dissertation, Harvard University, Cambridge, USA, 2000.

[12] μ-Adaptive Multi-domain Power aware Sensors: `http://www-mtl.mit.edu/research/icsystems/uamps`, April, 2001.

[13] 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*, satellite workshop of *16th Annual International Parallel & Distributed Processing Symposium* – IPDPS'02.

[14] K. Mehlhorn and S. Näher: LEDA: A Platform for Combinatorial and Geometric Computing. *Cambridge University Press*, 1999.

[15] C.E. Perkins: Ad Hoc Networking. *Addison-Wesley*, Boston, USA, January, 2001.

[16] S. Singh and C.S. Raghavendra: PAMAS: Power aware multi-access protocol with signalling for ad hoc networks. In *ACM Computer Communication Review*, vol. 28, no. 3, pp. 5-25, July 1998.

[17] C. Schurgers, V. Tsiatis, S. Ganeriwal and M. Srivastava: Topology Management for Sensor Networks: Exploiting Latency and Density. In *3rd ACM Annual International Symposium on Mobile Ad hoc Networks and Communication*, pp. 135-145, June 2002.

[18] TinyOS: A Component-based OS for the Network Sensor Regime. `http://webs.cs.berkeley.edu/tos/`, October, 2002.

[19] 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.