Energy Efficient Protocols for Sensing Multiple Events in Smart Dust Networks[1]

S. Nikoletseas, I. Chatzigiannakis,
H. Euthimiou, A. Kinalis
Computer Technology Institute
P.O. Box 1122, 26110 Patras, Greece
and Department of Computer
Engineering and Informatics
University of Patras, 26500 Patras, Greece
{*nikole, ichatz*}*@cti.gr,*
{*euthimio, kinalis*}*@ceid.upatras.gr*

A. Antoniou, G. Mylonas
Department of Computer
Engineering and Informatics
University of Patras, 26500 Patras, Greece
{*antonioa, mylonasg*}*@ceid.upatras.gr*

Abstract

*Wireless Sensor Networks are comprised of a vast number of ultra-small, autonomous computing and communication devices, with restricted energy, that co-operate to accomplish a large sensing task. In this work: a) We propose extended versions of two data propagation protocols for such networks: the Sleep-Awake Probabilistic Forwarding Protocol (***SW-PFR***) and the Hierarchical Threshold sensitive Energy Efficient Network protocol (***H-TEEN***). These non-trivial extensions improve 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, b) We implemented the two protocols and performed an extensive simulation comparison of various important measures of their performance with a focus on energy consumption, c) We investigate in detail the relative advantages and disadvantages of each protocol , d) We discuss a possible hybrid combination of the two protocols towards optimizing certain goals.*

1. Introduction

Recent dramatic developments in micro-electro-mechanical (MEMS) systems, wireless communications and digital electronics have already led to the development of small in size, low-power, low-cost sensor devices. Such extremely small devices integrate sensing, data processing and communication capabilities. Large numbers of sensor nodes can be deployed in areas of interest (such as inaccessible terrains or disaster places) and use *self-organization and collaborative methods* to form a sensor network.

Their wide range of applications is based on the possible use of various sensor types (i.e. thermal, visual, seismic, acoustic, radar, magnetic, etc.) in order to monitor a wide variety of conditions (e.g. temperature, object presence and movement, humidity, pressure, noise levels etc.). Thus, sensor networks can be used for continuous sensing, event detection, location sensing as well as micro-sensing. Hence, sensor networks have important applications, including (a) military (like forces and equipment monitoring, battlefield surveillance, targeting, nuclear, biological and chemical attack detection), (b) environmental applications (such as fire detection, flood detection, precision agriculture), (c) health applications (like telemonitoring of human physiological data) and (d) home applications (e.g. smart environments and home automation). For an excellent survey of wireless sensor networks see [1].

Note however that the efficient and robust realization of such large, highly-dynamic, complex, non-conventional networking environments is *a challenging algorithmic and technological task.* Features including the huge number of sensor devices involved, the severe power, computational and memory limitations, their dense deployment and frequent failures, pose *new design and implementation aspects* which are essentially different not only with respect to distributed computing and systems approaches but also to adhoc networking techniques.

[1]This work has been partially supported by the IST Programme of the European Union under contract numbers IST-1999-14186 (**ALCOM-FT**), IST-2001-33116 (**FLAGS**) and IST-2001-33135 (**CRESCCO**). Contact Author: S. Nikoletseas (nikole@cti.gr).

1.1. Our Contribution

We study the problem of *multiple event detection and propagation*, i.e. the local sensing of a series of crucial events and the energy efficient propagation of data reporting the realization of these events to a (fixed or mobile) control center. The control center could in fact be some human authorities responsible of taking action upon the realization of the crucial event. We use the term *"sink"* for this control center. We note that this problem *generalizes* the single event propagation problem (w.r.t. to [2, 4–6]) and poses new challenges for designing efficient data propagation protocols.

Under these more general and realistic (in terms of motivation by practice applications) modelling assumptions, we have implemented and experimentally evaluated, two information propagation protocols: (a) The Sleep-Awake Probabilistic Forwarding Protocol (**SW-PFR**) that avoids flooding by favoring in a probabilistic way certain "close to optimal" data transmissions and also allows particles to alternate between sleeping and awake modes to save energy, (b) the hierarchical version of the Threshold sensitive Energy Efficient Network Protocol (**H-TEEN**), where particles self-organize into clusters and build a tree of transmissions, propagating data only to their parent (cluster-head) in this tree. We note that we had to carefully design the protocols to work under the new network models.

We propose, implement and evaluate using simulation a modified version of the **TEEN** protocol introduced in [11]. In particular, we extended the concept of clustering used in the original **TEEN** protocol to that of *hierarchical* clustering. This possibility was discussed in the original paper. We show that in large area networks and when the number of layers in the hierarchy is small, **TEEN** tends to consume a lot of energy, because of long distance transmissions. On the other hand, when the number of layers increases, the transmissions become shorter, however there is a significant overhead in the setup phase as well as the operation of the network. In the light of the above, we chose to implement a *4-layer* hierarchical clustering. The main idea behind the hierarchical clustering is to enable the **TEEN** protocol to operate in networks that occupy large areas.

As opposed to [3] (where a 2-dimensional *lattice* deployment of smart dust particles has been used) *we here extend the network model* to the general case of particle deployment according to a random, uniform distribution. We further extend the **PFR** protocol and the network model by enabling the smart dust particles to periodically enter a power save mode, which we call the *sleeping mode*. Finally, we implement **PFR** so as to efficiently handle not only a single event, as described in [3] [2], but to propagate *multiple events*.

The extensive simulations we have performed show that both protocols are successful. In the setting we considered, **SW-PFR** seems *more robust in the case of high rates of event generation and large networks* (considering the area they occupy), while **H-TEEN** seems to perform better in smaller networks. We also concluded that the values of certain parameters, such as the radius of transmission, are highly critical for the performance of the **SW-PFR** protocol, given the network area and the number of the particles spread, and we investigate the precise effect of such parameters on performance.

In the light of the above, we also propose and discuss a *hybrid protocol* combining the two protocols and an intermediate randomization switching phase between them. This hybrid protocol can be parameterized towards achieving certain desired goals and trade-offs.

1.2. Discussion of Selected Related Work

A family of negotiation-based information dissemination protocols suitable for wireless sensor networks is presented in [8].

A data dissemination paradigm called *directed diffusion* for sensor networks is presented in [10], where data-generated by sensor nodes is named by attribute-value pairs.

In [7] a clustering-based protocol is given that utilizes randomized rotation of local cluster heads to evenly distribute the energy load among the sensors in the network. In [11] a new energy efficient routing protocol is introduced 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.

2. The Model

Sensor networks are comprised of a vast number of ultra-small homogenous sensors, which we here call *"grain" particles*. Each 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 (in our model here) cannot move. 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 [7–11].

Definition 1 *Let n be the number of smart dust particles and let d (usually measured in numbers of $particles/m^2$) be the* density *of particles in the area.*

There is a single point in the network area, which we call the sink $\mathcal{S}$, and represents a control center where data should be propagated to. Furthermore, we assume that there is a set-up phase of the smart dust network, during which the smart dust 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 $\mathcal{S}$. By assuming that each smart-dust particle has individually *a sense of direction*, and using these control messages, each particle is aware of the general location of $\mathcal{S}$.

Each particle is equipped with a set of monitors (sensors) for light, pressure, humidity, temperature etc. Each particle may have two communication modes: a *broadcast* (digital radio) *beacon mode* which can be also a directed transmission of angle α around a certain line (possibly using some special kind of antenna, see fig. 2) and a *directed to a point* data transmission mode (usually via a laser beam). Both the transmission range (which we denote by R) and the transmission angle (let it be α) can vary. Note that the protocols we study in this work can operate even under the broadcast communication mode. The laser possibility is added for reducing energy dissipation in long distance transmissions.

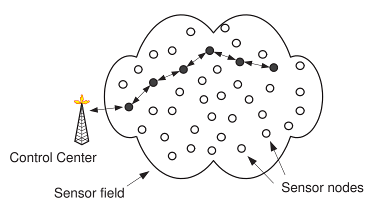

Figure 1. A Smart Dust Cloud

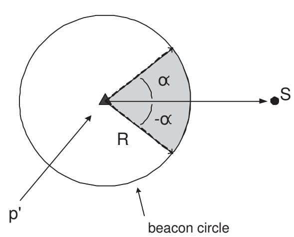

Figure 2. Directed transmission of angle α

Each particle can be in one of four different modes at any given time, regarding the energy consumption. These modes are: (a) transmission of a message, (b) reception of a message, (c) sensing of events and (d) sleeping. During the sleeping mode, particle ceases any communication with the environment, thus it is unable to receive any message or sense an event. In our model, we assume that the energy consumption of a sleeping particle is negligible, but it needs a certain amount of energy to return to the sensing state.

Following [7], for the case of transmitting and receiving a message we assume the following simple model where the radio dissipates E_{elec} to run the transmitter and receiver circuitry and ϵ_{amp} for the transmit amplifier to achieve acceptable SNR (signal to noise ratio). We also assume an r^2 energy consumption due to channel transmission at distance r. Thus to transmit a `k-bit` message at distance r in our model, the radio expends

$$E_T(k,r) = E_{T-elec}(k) + E_{T-amp}(k,r)$$
$$E_T(k,r) = E_{elec} \cdot k + \epsilon_{amp} \cdot k \cdot r^2$$

and to receive this message, the radio expends

$$E_R(k) = E_{R-elec}(k)$$
$$E_R(k,r) = E_{elec} \cdot k$$

where E_{T-elec}, E_{R-elec} stand for the energy consumed by the transmitter's and receiver's electronics, respectively.

Concluding, there are four different kinds of energy dissipation which are:

- E_T: Energy dissipation for transmission.
- E_R: Energy dissipation for receiving.
- E_{idle}: Energy dissipation for idle state.
- $E_{powerup}$: Energy dissipation for returning from sleeping state.

For the idle state, we assume that the energy consumed for the circuity is constant for each time unit and equals E_{elec} (the time unit is 1 second). On the other hand, the power up energy equals three times the amount of energy consumed in a time unit during the idle state, that is $3 \cdot E_{elec}$.

The particles in the **H-TEEN** protocol do not exploit the ability to get to sleeping mode, thus in that case the energy dissipation is characterized by the rest three types of energy consumption.

Finally, we assume that each particle, in protocol **H-TEEN**, does not spend any amount of energy to listen what the other nodes, of its cluster, send to the cluster head.

We note that in our simulations we *explicitly measure the above energy costs*. Our model, although simple, depicts accurately enough the technological specifications of real smart dust systems. In contrast to [10], 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.

3. The Problem

Assume the realization of a series of K crucial events $\mathcal{E}_i$, with each event being sensed by a single particle p_i ($i = 1, 2, \ldots, K$). Then the *multiple event propagation problem* $\mathcal{P}$ is the following:

"How can each particle p_i $(i = 1, 2, \ldots, K)$, via cooperation with the rest of the grain particles, efficiently (mainly with respect to energy) propagate information *info($\mathcal{E}_i$)* reporting realization of event $\mathcal{E}_i$ to the sink $\mathcal{S}$?"

We remark that this problem is a *generalization* of the single event propagation problem, which is more difficult to cope with.

Certainly, because of the dense deployment of sensor particles close to each other, communication between two particles is much more energy efficient than direct transmission to the sink. Furthermore, short-range hop-by-hop transmissions can effectively overcome some of the signal propagation effects in long-distance transmissions and may help to smoothly adjust propagation around obstacles. Finally, the low energy transmission in multi-hop communication may enhance security, protecting from undesired discovery of the data propagation operation.

On the other hand, long-range transmissions require the participation of few particles and therefore reduce the overhead on particle resources and provide better network response times. Furthermore, long-range communication permits the deployment of clustering and other efficient techniques, developed for ad-hoc wireless networks. In particular, a clustering scheme enables cluster heads to reduce the amount of transmitted data by aggregating information.

The above suggest that many diverse approaches exist to the solution of the *multiple event propagation problem* $\mathcal{P}$. In any case, the objective is to propagate most, if not all, of the events $\mathcal{E}_i$ to the sink efficiently. Note that, because of the multiplicity of events sensed and the small energy supplies, it is not obvious at all that all events sensed manage to be reported to the sink. Efficiency means to minimize the energy consumption in the sensor network, either by minimizing the number of transmissions and also minimize duplicate messages, observed in multi-hop delivery, or by reducing data size (by aggregations techniques) and the number of transmissions (by clustering) in long-range communication. Note that an appropriate MAC protocol is required in either case to handle collisions and avoid message retransmissions [13].

Furthermore, an interesting aspect of the problem under investigation is the lifetime of particles, since it affects the ability of the network to propagate data to the sink, because available routes are reduced as more particles consume their energy resources and "die".

4. The Sleep-Awake Probabilistic Forwarding Protocol (SW-PFR)

The basic idea of the protocol lies in probabilistically favouring transmissions towards the sink within a *thin zone* of particles around the line connecting the particle sensing the event $\mathcal{E}$ and the sink (see Fig. 3 for graphical representation). Although data propagation along this line is optimal with respect to energy and time cost, such propagation is not always feasible. This is true because, even if initially this direct line was appropriately occupied by sensors, certain sensors on this line might become inactive, either permanently (because their energy has been exhausted) or temporarily (because these sensors might enter a sleeping mode to save energy).

The protocol evolves in two phases:

Phase 1: The "Front" Creation Phase. During this initial phase a sufficiently large front is built to ensure that the data propagation process survives for an appropriately large period of time. This front is created by using a "flooding" mechanism for a configurable number of steps. According to this mechanism, the header of each message includes a counter called β. This counter is set to a predefined value by the source particle, when the latter generates the message relative to the sensed event. Following this initialization, each particle, upon receiving a pertinent message containing a positive β counter, reduces its value by 1 and *deterministically* forwards the message towards the sink. In order to do so, each particle uses directed "angle" transmission to broadcast data to all of its neighbors, that lie in the direction of the sink. When the β counter becomes zero, the particle proceeds to the second phase of the **SW-PFR** protocol. Ultimately, the β counter determines the length of the first phase of the **SW-PFR** protocol.

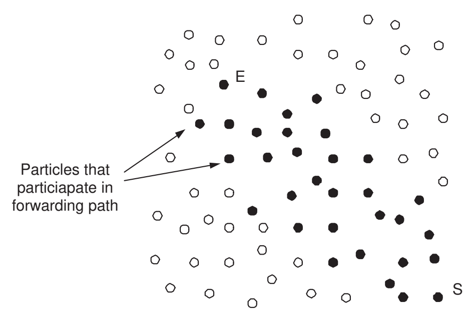

Figure 3. Thin zone of particles around the line connecting the particle sensing the event $\mathcal{E}$ and the sink $\mathcal{S}$.

Phase 2: The Probabilistic Forwarding Phase. In this second phase, data propagation is done in a probabilistic manner. Each particle calculates a probability of participation in the propagation process. The closer a particle is to the optimal transmission line, connecting the source node $\mathcal{E}$ detecting an event and the sink $\mathcal{S}$, the higher its probability to forward data pertinent to that particular sense event.

The "forwarding probability" $\mathbb{P}_{fwd}$ is chosen to be

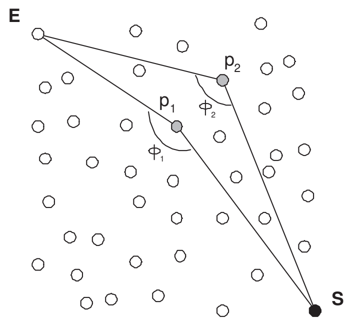

Figure 4. Angle ϕ and closeness to optimal line

$\mathbb{P}_{fwd} = \frac{\phi}{\pi}$ where ϕ is the angle defined by (a) the line connecting the particle performing the random choice and the sensor that initially sensed the event and (b) the line connecting this node to the sink. Remark that indeed *a bigger angle ϕ suggests a sensor position closer to the direct line between $\mathcal{E}$ and $\mathcal{S}$*, Figure 4 displays this graphically. Clearly, when $\phi = \pi$, then the sensor lies on this line. Thus, we get that $\phi_1 > \phi_2$ implies that for the corresponding particles p_1, p_2, p_1 is closer to the $\mathcal{E}$-$\mathcal{S}$ line than p_2, thus $\mathbb{P}_{fwd}(p_1) = \frac{\phi_1}{\pi} > \frac{\phi_2}{\pi} = \mathbb{P}_{fwd}(p_2)$

Angle ϕ calculation. Under appropriate (and realistic) modelling assumptions for sensor particles, angle ϕ can be locally calculated running a simple subprotocol (see [3] for detailed discussion). Such modelling assumptions include a) the ability of sensor particles to estimate the direction of a received transmission (e.g. by direction-sensing antennae), b) to estimate the distance from a nearby particle that did the transmission (e.g. via signal attenuation estimation techniques), c) to have a common a common co-ordinates system and d) to know the direction towards the sink (this is possibly done during a set-up phase). Note that we do not need GPS information or global network structure knowledge.

The **SW-PFR** protocol proposed in this paper is actually a non-trivial modification of the **PFR** protocol proposed in [3]. There are two major additions made to that protocol. First, the current version of **PFR**, **SW-PFR**, targets sensor-networks with multiple event generations. In particular, at any given time there could be more than one event being propagated towards the sink. In order to avoid repeated transmissions and infinite loops, each particle is provided with a limited "cache memory". In this cache, the particle registers the event IDs for each distinct event it has "heard of". Each event ID's uniqueness is guaranteed, by choosing it to be a concatenation of the source particle ID and the timestamp of the sensed event. Upon the receival of a message, a particle checks whether the pertinent event is enlisted in its cache. If that event is not in the particle's cache, it is registered and then the particle proceeds to the proper actions defined by the **PFR** protocol. However, if the event was already seen, the message is dropped and no further action is taken.

Presumably, a relatively small amount of memory (e.g. up to 2MB) would be adequate for such purpose. Note, that in the future the particle cache could enforce a policy of limited lifetime for each of its contents, thus reducing the space requirements to a minimum. Data aggregation also poses a challenge for further study and efficiency assessment.

As a second modification, this version of **PFR**, **SW-PFR**, encompasses an intriguing sleep-awake scheme. According to this scheme, each particle goes through alternating periods of "sleeping" and "awake". During a sleeping period, the particles cease any communication with the environment, thus the power consumption is assumed to be minimal and practically insignificant, whereas when a particle is awake, it consumes the regular amount of energy. In addition, a special energy amount is considered to be spent during the transition from "sleep" to "awake".

We assume that the sleeping/awake time periods alternate stochastically independently in each particle and have durations s, w respectively.

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

Definition 2 *The energy saving specification is:* $en = \frac{s}{s+w} = 1 - \frac{1}{1+\gamma}$

5. A Hierarchical Threshold-sensitive Energy Efficient Network Protocol (H-TEEN)

The Threshold sensitive Energy Efficient Network protocol (**TEEN**) was introduced in [11] and is essentially an interesting modification of the fundamental **LEACH** protocol (Low-Energy Adaptive Clustering Hierarchy, [7]) for reactive sensor networks. By reactive sensor networks we mean networks where particles immediately react only to sudden and drastic changes in the value of a sensed attribute. The basic concepts in **TEEN** are the clustering of particles and the use of thresholds in order to decide whether a particle should transmit data to the sink.

TEEN uses particle self-organization into clusters, originally proposed in **LEACH**, in order to reduce transmissions. We have extended the concept of clustering to a *hierarchical* clustering, a proposal also made in the original paper. A tree of transmissions is built, where each particle transmits data only to its parent (cluster-head) in this tree. To be more specific, **TEEN**'s operation is divided into rounds, and every round is further divided into a short set-up phase, where the organization in clusters occurs, and a relatively long steady-phase which is the phase where normal network operation occurs.

Setup phase: In this phase, each particle decides on whether it should become a cluster-head or not. This decision is based on a fixed probability $\mathbb{P}_c$ (in our simulations $\mathbb{P}_c = 0.05$) and on whether it has been a cluster head in the last $1/\mathbb{P}_c$ rounds. Specifically, a particle n picks a random number from 0 to 1 and compares it to a threshold $T(n)$, which is calculated in every round as

$$T(n) = \begin{cases} \frac{\mathbb{P}_c}{1-\mathbb{P}_c*(r \bmod \frac{1}{\mathbb{P}_c})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

Here r is the current round and G is the set of particles that have not been elected as cluster-heads in the last $1/\mathbb{P}_c$ rounds. As we can see, this threshold is growing as rounds pass and so we can ensure that every node alive will become a cluster-head at some point within $1/\mathbb{P}_c$ rounds. The same process begins over after $1/\mathbb{P}_c$ rounds. After a particle decides to become a cluster-head, it broadcasts an advertisement message to the entire network. The other particles hear the advertisements from all the cluster-heads, and choose the cluster-head they belong to based on the strength of the signal they received.

When a particle decides on which cluster it wants to belong to, it transmits a response back to the cluster-head of the corresponding cluster. This is probably done using some kind of `CSMA MAC` protocol to avoid collisions. The cluster-head at some point decides that it has heard all of the responses from the particles in its cluster using a time-out and then sets up a `TDMA` schedule for transmissions. It then broadcasts this schedule to the particles in its cluster.

At this time, the next level of hierarchy can be built. We extended the original **TEEN** paper with the following scheme for hierarchical clustering. The nodes that have become cluster-heads, decide on whether they should pass on the next level of hierarchy with the same probability $\mathbb{P}_h$ (in our simulations $\mathbb{P}_h = 0.1$) and on whether the number of cluster-heads advertisements in the previous level of hierarchy received is less than eight. If a particle decides on moving to the next level of hierarchy it broadcasts an advertisement message again and only cluster-heads of the previous level decide to which cluster-head of the next level they belong to, and this process is repeated for the next levels. In our simulation we have used four levels of hierarchy.

Steady phase: After all the levels of clustering have been set up, the actual data transmissions from particles can begin. The transmissions from particles to cluster-heads are done according to the `TDMA` schedule of each cluster. To reduce interference between different neighboring clusters, any cluster-head picks randomly a code from a certain list and so particles in each cluster use different `CDMA` codes. Finally, there are two thresholds, the “hard” and the “soft” threshold, which determine when a particle must transmit data to the base station, since particles sense continuously the field. The first one is a value beyond which the particle enters an alarm state. The second one is a small change in the value of the sensed attribute that took place in the field.

By the first time hard threshold is reached, the particle stores the sensed value and transmits it to the base station. If the next sample differs from the stored value by at least the soft threshold, this information should be transmitted back to the base station. An example of these thresholds is when the measured attribute is temperature; we set the hard threshold to be 30^o and the soft threshold any change over 0.5^o. In our simulations we used the notion of an event, meaning a situation where particles must report to the base station, rather than just random values of the sensed attributes. So, when an event occurs, it is implied that these thresholds are outreached.

6. Implementation Details

For the purpose of the comparative study of the previously described protocols, we have designed a new simulator, which we named **simDust**, that was implemented in `Linux` using `C++` and the `LEDA` [12] algorithmic and data structures library.

Another interesting feature of our simulator, is the ability to experiment with very large networks. In fact, the complexity of extending existing networks simulators, and their (in cases of large instances) time consuming execution, were two major reasons for creating this simulator. **simDust** enables the protocol designer to implement the protocol using just `C++` and avoids complicate procedures that involve the use of more than one programming language.

Additionally, **simDust** generates all the necessary statistics at the end of each simulation based on a big variety of metrics that are implemented (such as delivery percentage, energy consumption, delivery delay, longlivety, etc.)

The key points in **simDust**'s implementation are the following:

Operation in rounds: A basic concept used in the simulator is that its operation is divided into discrete rounds, both in **SW-PFR** and **H-TEEN**. One round represents a time interval in which a particle can transmit or receive a message and process it according to the protocol that is being simulated.

`MAC` layer assumptions: **simDust** leaves transmission collisions to be handled by lower `MAC` layer protocols and does not take them into account. It is our intention to consider them in next versions of this simulator.

Energy assumptions: We have included an energy dissipation scheme for both protocols implemented. In particular, we have assumed that a particle consumes a standard amount of energy E_{elec} per round while being awake. Furthermore, in each transmission energy fading is proportional to the square of the distance. For each receive, a node is credited with an amount of energy that practically reflects the power needed to run the transceiver circuit namely E_{elec}. Finally, a particle can switch over to the sleep state, to save energy. No energy consumption virtually takes place while the particle remains in the sleep mode, since it keeps its transceiver and its sensors shut down.

Size of messages: Regarding the communication cost in terms of the bits transmitted per message, we assume that information messages require 1Kbyte, plus a 40 bits header, containing a 32 bit identifier for the sender particle and an 8 bit code that determines the message type.

SW-PFR specific assumptions: We assume that the protocol's execution is preceded by an initialization phase, where each particle discovers all the neighboring particles that are both within transmission range and within the cycle sector demarcated by a predefined angle value. The transmission range of each particle's transmitter is set to be fixed and preferably of a value significantly smaller than the network diameter. The sensed events that can trigger a data propagation in the network can occur at any given round. Additionally, in the process of embedding a sleep-awake scheme in the **SW-PFR** protocol, we have used two configurable variables that effectively define the ratio between the duration of the sleep and awake periods. Finally, the required memory where each particle stores the events it has previously seen is considered to be of infinite capacity for reasons of keeping the implementation complexity considerably low.

H-TEEN specific assumptions: Our hierarchical implementation uses four levels of hierarchy. We used a relatively large interval of 24 rounds between successive cluster set-up rounds. Finally, we have calculated the size of the transmission schedule that is broadcast by each cluster-head as a header of 40 bits followed by 32 bits (size of particle ID) times the number of particles included in that particular cluster.

7. Efficiency Measures

On each execution of the experiment, let K be the total number of crucial events $(\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_K)$ and k the number of events that were *successfully* reported to the sink $\mathcal{S}$. Then, we define the *success rate* as follows:

Definition 3 The success rate, $\mathbb{P}_s$, is the fraction of the number of events *successfully* propagated to the sink over the *total number of events*, i.e. $\mathbb{P}_s = \frac{k}{K}$.

Another crucial efficiency measure is the average available energy of each particle in the network over time:

Definition 4 Let E_i be the available energy for the particle i. Then $E_{avg} = \frac{\sum_i^n E_i}{n}$ is the average energy per particle in the smart dust network, where n is the number of the total particles dropped.

Finally, we consider the number of alive particles as a measure of efficiency and the network survivability in each case. As in case of the energy, the more particles are alive the better. A source of valuable information is also the particular manner that particles die over time.

Definition 5 Let h_A (for "active") be the *number of "active" sensor particles* participating in the sensor network.

We also give another definition concerning the way the events are generated in the network. This parameter critically influence the efficiency measures defined above.

Definition 6 Let I_s be the *injection rate*, measured as the probability of occurrence of a crucial event during a round.

8. Simulation Results

We evaluate the performance of the two protocols by conducting a comparative experimental study. In our experiments, we generate a variety of sensor fields. The field size ranges from 200m by 200m to 1500m by 1500m. We note that these network sizes studies are significantly larger than those usually investigated in the relevant research and enable a study of the scalability of the network. In these fields, we drop $n \in [500, 3000]$ particles randomly uniformly distributed on the smart-dust plane.

We start by examining the success rate of the protocols with respect to the network size, for two different injection rates I_s (0.05 and 0.8). We focus on two *extreme* values to investigate the divergent protocol behavior in corresponding extreme settings

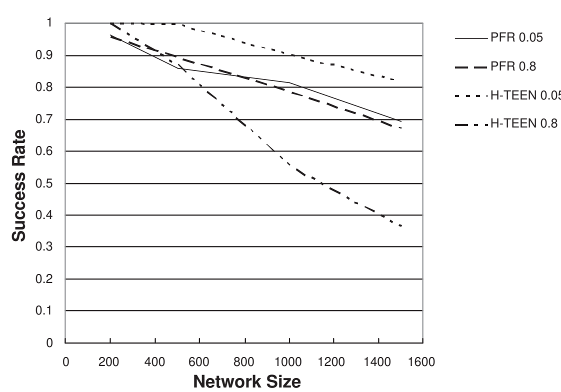

Figure 5. Success rate for various network sizes for injection rates 0.05 and 0.8.

In figure 5 the success rate of the protocols is depicted as the network size increases. It is clear that for small dimensions (200mx200m, 500mx500m) both protocols achieve high performance ($\geq$ 0.85), while performance drops as networks size increases. However, **SW-PFR**'s success rate remains quite high (i.e. ≥ 0.7) even for very large networks (i.e. 1500mx1500m), while **SW-PFR** seems to behave well in both the case of low and high event generation rates. On the contrary, **H-TEEN** seems to perform poorly in the case of frequent events since its success rate drops bellow 0.3 in that case.

We continue, by examining the success rate of the protocols with respect to the I_s for two different network sizes (Figure 6). Initially, for low injection rates, in small networks (500mx500m), both protocols behave almost optimally achieving a success rate close to one and decrease as injection rate increases. However, for larger network size the impact of the injection rate seems to be more significant. In particular, **H-TEEN**'s success rate drops from 85% to almost 30%, while **PFR**, even though its initial success rate is about 70%, seems to be less affected by the increase in injection rate.

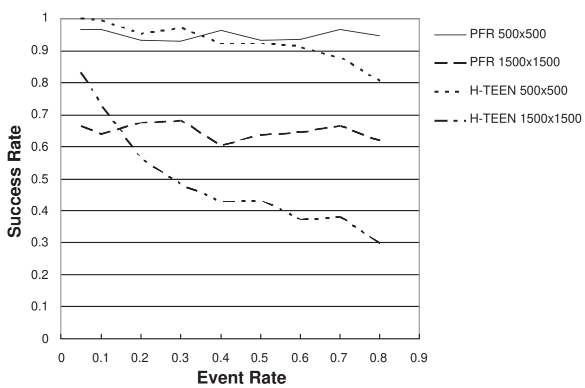

Figure 6. Success rate for various injection rates and network area 500x500, 1500x1500.

The apparent dependence of **TEEN** protocol's performance from the injection rate is due to its clustering scheme. As injection rate increases a cluster head is responsible for delivering more events, thus it consumes more energy during its leadership. If injection rate becomes too high, cluster heads are more likely to exhaust their energy supplies before a new cluster head is elected. When a cluster head "dies", the cluster ceases to function and all events that occur on that cluster are lost until a new cluster head is elected. Furthermore large network sizes worsen this phenomenon because more energy is required from the cluster head to propagate an event to the sink (since the energy spent in one hop is of the order of the square of the transmission distance).

On the other hand the **SW-PFR** protocol is mostly unaffected by high injection rates but influenced by larger network sizes. This is due to its multihop nature, since more hops are required for the propagation of an event.

We move on by examining the average energy consumed per particle in time. We remind the reader that each particle dropped to the smart-dust plane has 1Joule of energy to its disposal. Figure 7 depicts the average amount of energy that each particle has available in each round, for network areas of (500mx500m) and (1000mx1000m) where the injection rate $I_s = 0.05$. Note that in both cases the Success Rate, $\mathbb{P}_s$, is over 80%. For the case of the small network **H-TEEN** consumes less energy than **SW-PFR**, while on the larger network, **SW-PFR** seems less expensive.

To explain this phenomenon we have to consider the nature of the event propagation of the two protocols. The **H-TEEN** protocol uses a constant number of hops to propagate the messages, having its cluster heads transmitting directly to the next level of the hierarchy, while **SW-PFR** uses short-ranged multihop transmissions to deliver the message to the sink. When the network area is fairly small the transmissions of cluster heads cost less than the multiple multihop transmissions of **SW-PFR**. However, when the network area increases, the cluster heads have also to increase its transmitting power by an order of R^2, as the radio range R increases, in contrast to **SW-PFR** where the multihop transmissions increase linearly. So, for the case of large area networks **SW-PFR** seems more appropriate than **H-TEEN**.

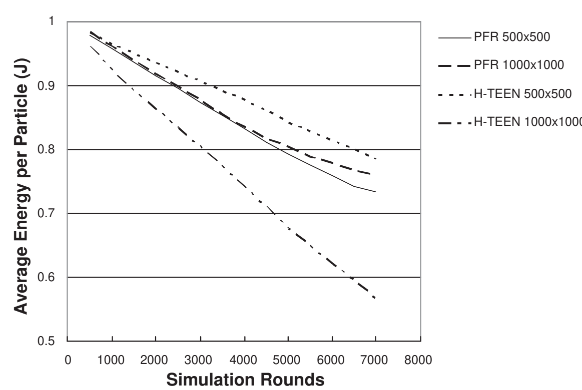

Figure 7. Average Energy per particle for $I_s = 0.05$ and network size 500x500, 1000x1000.

Finally, we examine the way the number of alive particles varies with time. In Figure 8 the number of alive particles is presented for network sizes 500m by 500m and 1000m by 1000m, for injection rate $I_s = 0.05$. Note that more particles are dropped in a 1000m by 1000m network, the more or less deal with similar densities in both protocols

We notice that for the **H-TEEN** protocol for both network sizes the number of alive nodes decreases at a constant rate, whilst for the **SW-PFR** protocol there is a sudden decrease in the number of alive particles. This observation depicts the property of the **H-TEEN** protocol to evenly distribute the energy dissipation to all the particles in the network. On the contrary, the **SW-PFR** protocol stresses more the particles which are placed closer to the sink, so at a point in time these particles start to "die" rapidly.

On the other hand, in the larger network area (1000m x 1000m) the particles in **H-TEEN** protocol "die" more rapidly than those of **SW-PFR**. This was expected because in **H-TEEN** protocol particles are forced to transmit in larger distances than those in **SW-PFR**, so they consume more energy and "die" faster. We should also notice the same behavior of **SW-PFR** as in fig. 8, from a point in time and on, particles start to die faster than before and this is because of the fact that **SW-PFR** stresses more the particles that lie near the sink forcing them to propagate the majority of the messages that occur in the smart-dust network.

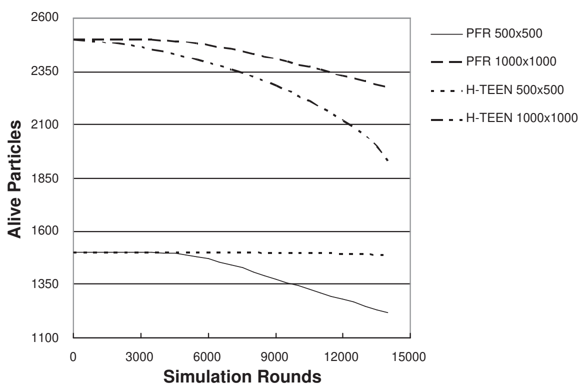

Figure 8. Number of alive particles for $I_s = 0.05$ and network size 500x500, 1000x1000.

For the **SW-PFR** protocol we adjusted some extra parameters such as the radio transmission range (R) and sleep-awake ratio (en), in order to achieve good performance for a fixed network density, while, the values of the parameters defined for the **H-TEEN** protocol are set as described in section 6. The choice of values for the **SW-PFR** protocol is very crucial for its performance (thus **SW-PFR** seems to more "programmable") while the **H-TEEN** protocol seems to less benefit from adjusting its parameters. In Figure 9 we give an example of how the success rate of the protocol can vary for different values of the radio range (R) in a 500m by 500m network with 1500 particles. We notice that there is a strict area of the radio range where the protocol behaves well, while outside this area its performance drops quickly. Thus, we provide an indication of how to exactly adjust the radio range to get high success rate (i.e. for success rate at least 0.8, R should be between 35 and 60 m, in the particular setting.)

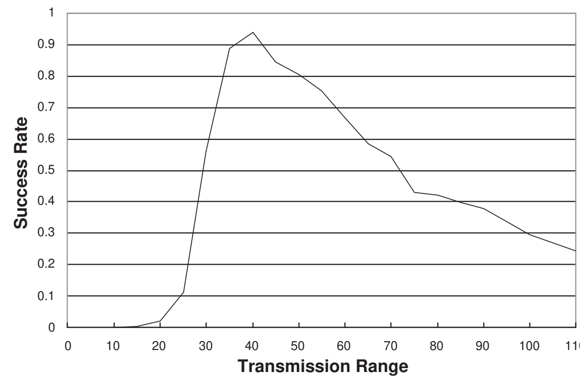

Figure 9. Success rate of SW-PFR for various values of radio range.

9. Towards a Hybrid Protocol

Overall, for the **H-TEEN** protocol the simulation suggests that in large area networks and when the number of layers in the hierarchy is small, **TEEN** tends to consume a lot of energy, because of long distance transmissions. On the other hand, when the number of layers increases, the transmissions become shorter, however there is a significant overhead in the setup phase as well as the operation of the network. Thus, **H-TEEN** exhibits a certain trade-off behavior with respect to the number, let it c, the numbers of layers used in the hierarchy. We here have implemented and evaluated the (rather intermediate) value c=4. We plan to further investigate, both by analytical and experimental means, this trade-off.

Furthermore, the main findings of this work are the following:

1. **SW-PFR** seems to be more energy efficient in networks covering large geographical areas. In such networks **H-TEEN** seems to be expensive.

2. On the other hand **H-TEEN** tends to be more efficient in small area networks and also it tends to somehow evenly distribute energy consumption among the particles locating the event.

In the light of the above two findings, we propose a possible *hybrid combination* of the protocols in large area networks. This hybrid protocol should have three distinct phases (we list below Phase 2 last in order, to better discuss how this intermediate phase works).

Phase 1: Initially, and for a sufficiently large part of the network area, **SW-PFR** is used to propagate data.

Phase 3: When data gets appropriately close to the sink, **H-TEEN** is employed.

Phase 2: Since **H-TEEN**'s energy consumption characteristics heavily depend on the spatial distribution of event occurrence, and since **H-TEEN** better performs in the case of rather uniform event placement, we introduce an intermediate phase between Phase 1 (**SW-PFR**) and Phase 3 (**H-TEEN**). This phase may use a "load balancing" technique, such as randomization, to rather uniformly further propagate data, in order to avoid worst case behavior for the **H-TEEN** protocol that would arise if data transmissions reach the **H-TEEN** employment area around the same, more or less, cluster heads. This randomization can be implemented in the following way: the particles in the "phase 2" application area select the next particle to propagate data by randomly and uniformly choosing the angle of transmission and the transmission radius, in some range. We note that the details of this hybrid protocols' design, i.e. how long each of the 3 phases should be, what is the range of the parameters (angle, radius) in the randomization phase etc, are quite complicated, so we plan to study them and implement and evaluate the hybrid protocol in a separate work.

10 Closing Remarks

We presented in this work two extended versions of the protocols PFR and TEEN, i.e. SW-PFR and H-TEEN, for information propagation in sensor networks. We have implemented the new protocols and conducted an extensive comparative experimental study on networks of large size to validate their performance and investigate their scalability. Our results basically show that the SW-PFR protocol achieves high success rates under specific conditions of network density and radio range (R), it is resistant to frequent events and operates efficiently in large area networks. On the other hand, the H-TEEN protocol achieves high success rates in small area networks but there is a deterioration of its performance when the size of the network or the injection rate increases.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Journal of Computer Networks*, Volume 38:393–422, 2002.

[2] 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 *Proceedings of the International Conference on Parallel and Distributed Computing–(EUROPAR)*, 2003. Distinguished Paper. Also accepted in the *Parallel Processing Letters (PPL) Journal*, to appear in 2004.

[3] I. Chatzigiannakis, T. Dimitriou, S. Nikoletseas, and P. Spirakis. A probabilistic forwarding protocol for efficient data propagation in sensor networks. In *5th European Wireless (EW) Conference*, 2004.

[4] I. Chatzigiannakis and S. Nikoletseas. A sleep-awake protocol for information propagation in smart dust networks. In *3rd Workshop on Mobile and Ad-Hoc Newtworks (WMAN), – IPDPS Workshops*. IEEE press, 2003.

[5] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. Smart dust protocols for local detection and propagation. In *2nd ACM Workshop on Principles of Mobile Computing – POMC*, 2002. Also accepted in the *ACM Mobile Networks (MONET) Journal*, to appear in 2004.

[6] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis. Efficient and robust protocols for local detection and propagation in smart dust networks. *ACM Mobile Networks (MONET) Journal, Special Issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks*, 2003.

[7] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy efficient communication protocol for wireless microsensor networks. In *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 *5th ACM/IEEE International Conference on Mobile Computing – MOBICOM*, 1999.

[9] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. Technical Report Technical Report 01-750, University of Southern California Computer Science Department, November 2001.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *6th ACM/IEEE International Conference on Mobile Computing – MOBICOM*, 2000.

[11] A. Manjeshwar and D. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. In *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*, 2002.

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

[13] J. H. W. Ye and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proc. 12th IEEE International Conference on Computer Networks – INFOCOM*, 2002.

[14] Wireless integrated sensor networks. `http:/www.janet.ucla.edu/WINS/`, April 2001.

[15] Tinyos: A component-based os for the network sensor regime. `http://webs.cs.berkeley.edu/tos/`, October 2002.