A New Energy Efficient and Fault-tolerant Protocol for Data Propagation in Smart Dust Networks using Varying Transmission Range *

Thanasis Antoniou[†] Azzedine Boukerche[‡] Ioannis Chatzigiannakis[§]
George Mylonas[†] Sotiris Nikoletseas[†§]

October 23, 2003

Abstract

Smart Dust is a special case of wireless sensor networks, comprised of a vast number of ultra-small fully autonomous computing, communication and sensing devices, with very restricted energy and computing capabilities, that co-operate to accomplish a large sensing task. Smart Dust can be very useful in practice i.e. in the local detection of remote crucial events and the propagation of data reporting their realization to a control center.

In this work we propose a new energy efficient and fault tolerant protocol for data propagation in smart dust networks, the Variable Transmission Range Protocol (VTRP). The basic idea of data propagation in VTRP is the varying range of data transmissions, ie. we allow the transmission range to increase in various ways. Thus data propagation in our protocol exhibits high fault-tolerance (by bypassing obstacles or faulty sensors) and increases network lifetime (since critical sensors, ie. close to the control center are not overused). As far as we know, it is the first time varying transmission range is used.

We *implement* the protocol and perform an *extensive experimental evaluation and comparison to a representative protocol* (LTP) of several important performance measures with a focus on energy consumption. Our findings indeed demonstrate that our protocol achieves significant improvements in energy efficiency and network lifetime.

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. Examining each such device individually might appear to have small utility, however the effective *distributed co-ordination* of large numbers of such devices may lead to the efficient accomplishment of large sensing tasks. 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.

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

[†]Dept of Computer Engineering and Informatics, University of Patras, 26500 Patras, Greece. Emails: {antonioa,mylonasg}@ceid.upatras.gr

[‡]SITE, University of Ottawa, Canada, Email: boukerche@site.uottawa.ca

[§]Computer Technology Institute, P.O. Box 1122, 26110 Patras, Greece, Emails: {ichatz,nikole}@cti.gr

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] and also [7, 13].

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 ad-hoc networking techniques.

Contribution: In this work we focus on an important problem under a particular model of sensor networks that we present. More specifically, 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 and fault tolerant 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, 3, 5]) and poses new challenges for designing efficient and fault tolerant data propagation protocols. The new protocol we present here can also be used for the more general problem of data propagation in sensor networks (see [11]).

The basic innovation in our protocol is to vary the range of data transmissions. This feature aims at better performance, compared to typical fixed transmission range data propagation, in some rather frequently occurring situations like:

(a) The case of low densities of sensor particles. In such networks, fixed range protocols may trap in backtracking actions when no particles towards the sink are found. Our protocol, by increasing the transmission range, may find such particles and avoid extensive backtracking.

(b) Because of the possibility to increase transmission range, VTRP performs better in cases of obstacles or faulty/sleeping sensors. Also, it bypasses certain critical sensors (like those close to the sink) that tend to be overused, and thus prolongs the network lifetime.

To demonstrate the above properties of VTRP, we compare it to a typical fixed range protocol: the Local Target Protocol (LTP).

The ability of LTP to propagate information regarding the realization of a crucial event to the control center depends on the particle density of the network. The experiments conducted in [3] indicate that for low particle densities, LTP fails to propagate the messages to the control center (while for high particle densities the failure rate drops very fast to zero, i.e. the messages are almost always reported correctly). The new protocol that we propose in this paper successfully overcomes this problem by increasing the transmission range of the particles that fail to locate an active neighboring particle towards the sink. In fact, the experiments conducted in this paper (see section 8) demonstrate the superiority of VTRP over LTP even for sensor networks with very low particle densities.

Further note that this is the first time that the LTP protocol is evaluated under the setting of multiple events. Our findings indicate that LTP has a fundamental design flaw in this case, as the success of the propagation process heavily depends on the lifetime of the particles that are located around the control center. As soon as these particles exhaust their power supplies, the whole network becomes inoperable. Note that this design flaw that protocols for sensor networks are prone too was first reported in [9]. The new protocol that we present here successfully overcomes this problem by adjusting the transmission range of the particles as soon as the particles closer to the control center "die". Our experiments indicate that VTRP increases the ability of the network to report multiple events up to 100%, compared to LTP.

We propose four different mechanisms for varying the transmission range of the particles that aim at different types of smart dust networks regarding particles densities and energy saving criteria. Our experimental results show that VTRP can be easily modified to further improve its performance. Actually $\mathsf{VTRP_p}$ (where range is increased aggressively) and $\mathsf{VTRP_r}$ (that randomizes between the various range change functions towards a better average case performance) successfully propagate about 50% more events that the "basic" VTRP and almost 200% more events that the original LTP protocol.

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.

A family of negotiation-based information dissemination protocols suitable for wireless sensor networks is presented in [10]. 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 [11], 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. [12] presents an alternative approach that constructs a greedy incremental tree that is more energy-efficient and improves path sharing.

A different approach for propagating information to the sink is to use routing techniques similar to those used in mobile ad-hoc networks ([18]). In [9] 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 [16] a new energy efficient routing protocol is introduced that does not provide periodic data monitoring (as in [9]), 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 [9] achieves less energy consumption and response time. A data propagation protocol (PFR) that favors in a probabilistic way certain "close to optimal" transmissions (thus saving energy) has been introduced in [4]. A modified version of the PFR protocol [3] has been proposed and comparatively evaluated with [9, 16] in [6].

Furthermore, this work is related to previous research of [2, 3], where new local detection and propagation protocols (including LTP) are proposed, that are very energy and time efficient, as shown by a rigorous average case analysis performed in these works under certain simplifying assumptions.

2 The Model

Sensor networks are comprised of a vast number of ultra-small homogenous sensors, which we here call *"grain" particles* (see also [2, 3]). 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 [9–12, 16].

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 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 $\mathcal{S}$.

The particles are 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). In our model, we assume that the transmission range (R) can vary (i.e. by setting the transmission power at appropriate levels) while the transmission angle (let it be α) is fixed and cannot change throughout the operation of the network (since this would require a modification or movement of the antenna used). Note that the protocols we study in this work can operate even under the broadcast communication mode (i.e. $\alpha = 2\pi$). The laser possibility is added for reducing energy dissipation in long distance transmissions.

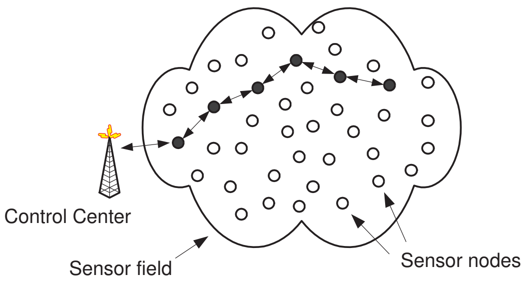


Figure 1: A Smart Dust Cloud

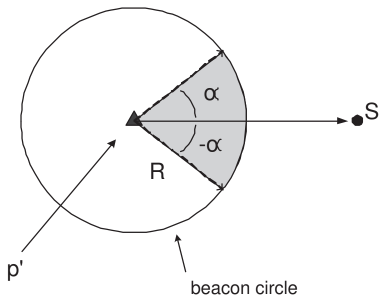


Figure 2: Directed transmission of angle α

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 and (c) sensing of events.

Following [9], 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.

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).

We note that in our simulations we *explicitly measure the above energy costs*. 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 [9, 16]). In contrast to [11, 14], 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 [8] a thorough comparative study and description of smart dust systems is given, from the technological point of view.

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, in an efficient (mainly with respect to energy and time) and fault-tolerant way, 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 because of the severe energy restrictions of the particles.

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* $\mathcal{P}$. Further to choosing between long or short transmissions, certain additional trade-offs are introduced by choosing between fixed or varying transmission range. In particular, we wish to focus on the following important properties:

(a) *Obstacle avoidance:* This may be achieved by increasing transmission range when an obstacle is encountered.

(b) *Fault tolerance:* Increasing range may reach active sensors when the current range does not succeed, either because of faulty or "sleeping" sensors close to sensor which is currently transmitting or in the case of very low network densities.

(c) *Network longevity:* 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". Varying transmission range may bypass the sensors lying close to the sink, that tend to be overused in case of fixed range transmissions, since all data passes through them in this case. The same holds also in the case of a geographical concentration of event generation.

4 The Variable Transmission Range Protocol (VTRP)

In this protocol, each particle p' that has received *info($\mathcal{E}$)* from p (via, possibly, other particles) does the following:

Phase 1: The Search Phase. It uses a periodic low energy broadcast of a beacon in order to discover a particle nearer to $\mathcal{S}$ than itself. Among the particles returned, p' selects a unique particle p'' that is "best" with respect to progress towards the sink. More specifically, the particle p''_E that among all particles found achieves the bigger progress on the $p'S$ line, should be selected (see Fig. 2).

Phase 2: The Direct Transmission Phase Then, p' sends *info($\mathcal{E}$)* to p'' and sends a *success* message to p (i.e. to the particle that it originally received the information from).

Phase 3: The Transmission Range Variation Phase. If the *search phase* fails to discover a particle nearer to $\mathcal{S}$, p' enters the *transmission range variation phase*. More specifically, each particle maintains a local counter τ, with initial value $\tau = 0$. Every time the *search phase* fails, this counter is increased by 1. Thus τ is an indication of the number of failures to locate an active particle. Based on τ, the particle modifies its transmission range $\mathcal{R}$ according to a change-function $\mathcal{F}(\tau)$. We here consider four different functions for varying the transmission range:

(a) *Constant Progress.* This choice is more suitable in the case where the network is comprised of a large number of particles and thus, a small increment of the transmission range will probably suffice to locate an active particle. Based on this assumption, the change-function is defined as follows:

$$\mathcal{F}(\tau) = \mathcal{R}_{new} = \mathcal{R}_{init} + c \cdot \tau \text{ , where } c \text{ is a constant set to a small value, i.e. } c = 10$$

This is considered as the "basis" VTRP and is denoted as $\mathsf{VTRP_c}$.

(b) *Multiplicative Progress.* In this case, the transmission range of the particle is increased mode drastically. We call this variation of our protocol $\mathsf{VTRP_m}$.

$$\mathcal{F}(\tau) = \mathcal{R}_{new} = \mathcal{R}_{init} + \mathcal{R}_{init} \cdot m \cdot \tau \text{ , where } m \text{ is a constant set to a small value, i.e. } m = 3$$

This drastic change has bigger probability of finding an active particle, however it leads to higher energy consumption.

(c) *Power Progress.* In this case, the transmission range of the particle is increased even faster using the following scheme:

$$\mathcal{F}(\tau) = \mathcal{R}_{new} = \mathcal{R}_{init} + \mathcal{R}_{init}^{\sqrt{(\tau+1)}}$$

We call this protocol $\mathsf{VTRP_p}$.

(d) *Random Progress.* When the density of the network is not known in advance, we use randomization to avoid bad behavior due to the worst case input distributions for each choice above (i.e. small modifications to the transmission range in $\mathsf{VTRP_c}$ in case of low densities and big modifications resulting from $\mathsf{VTRP_p}$ in high particle densities). We call this variation $\mathsf{VTRP_r}$ and is defined as follows:

$$\mathcal{F}(0) \; = \; \mathcal{R}_{init}$$

$$\mathcal{F}(\tau) \; = \; \mathcal{F}(\tau - 1) \; + \; \mathcal{R}_{init} \cdot r \; , \text{ where } r \in (0, 8], \text{ a random value}$$

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 VTRP 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.

5 The Local Target Protocol (LTP)

LTP, introduced in [3], is similar to VTRP except Phase 3, where a backtrack mechanism is implemented, instead of modifying the particle's transmission range in the case when no particles towards the sink are found. We make the assumption that information *info($\mathcal{E}$)* is generated at a particle p (when it detects an event) and it is transmitted to a particle p' using the first two phases. Every particle p', as stated above, maintains some information about the particle p from which *info($\mathcal{E}$)* was originally transmitted. We provide below LTP Phase 3 only.

Phase 3 in LTP: The Backtrack Phase. If Phase 1 fails several a certain (appropriately chosen) number of times, i.e. an awake neighbor particle p'' is not found in the particle's search area, then p' will send a failure notice and *info($\mathcal{E}$)* back to p. If p is the source of *info($\mathcal{E}$)*, then it will decide that propagation of this information towards the control center is impossible and erases *info($\mathcal{E}$)* from its memory.

6 Implementation Details

To implement the protocols presented in the previous sections, we have used simDust [6], that operates in `Linux` using `C++` and the `LEDA` [17] algorithmic and data structures library. An interesting feature for our simulator, is its ability to experiment with very large networks of thousands of nodes. 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 wide variety of metrics that are implemented (such as delivery percentage, energy consumption, delivery delay, longevity, 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. 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 a detailed 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 consumption is proportional to the square of the transmission 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 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.

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 2 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 of our comparative evaluation of the two protocols is the average available energy of each particle in the network over time:

Definition 3 Let E_i be the available energy for the particle i. Then $E_{tot} = \sum_i^n E_i$ is the total energy available in the smart dust network, where n is the number of the total particles dropped. Note that E_i and E_{tot} vary with time.

Clearly, the less energy a protocol consumes the better, but we have to notice that the comparison, in order to be fair, should be done in cases where the other parameters of efficiency should be similar (i.e. satisfy certain quality of service guarantees).

Finally, we consider as a measure of efficiency of the two protocols the number of alive particles, capturing the network survivability in each case. As in case of the energy, the more particles are alive the better. A source of crucial information is also the particular manner that particles die over time, such as the geographical distribution of the nodes that die out earlier, the evolution of energy consumption in critical sensors such as those lying close to control center.

Definition 4 Let h_A (for "alive") be the *number of "alive" sensor particles* participating in the sensor network.

8 Experimental Results

We start our experimentation by evaluating the effect of the particle density on the performance of the new protocol VTRP when compared to the already existing one, LTP. We generate a variety of sensor fields in a 2000m by 2000m square and in these fields, we drop $n \in [1000, 8000]$ particles uniformly distributed on the smart-dust plane. In each execution, *we generate a single event by randomly selecting a particle in the network*. The results of this experiments are shown in fig. 3.

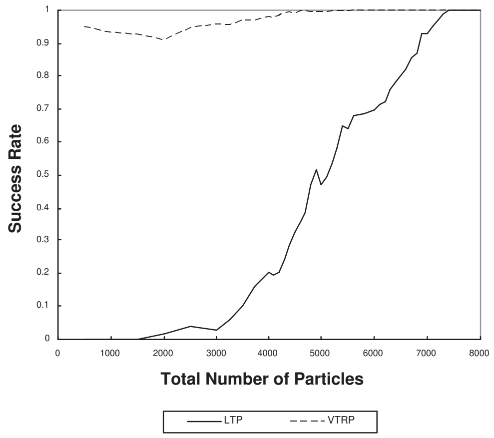


Figure 3: Success Rate ($\mathbb{P}_s$) for LTP and VTRP for various particle densities ($n \in [1000, 8000]$).

It is evident that the effect of particle density has significant impact on the performance of LTP. We observe that for low densities (i.e. $n \leq 2000$) the protocol almost always fails to report the event to $\mathcal{S}$, while when $n \geq 5000$ the success rate increases approaching very fast one. This can be justified by taking into account the average degree of each particle for various network sizes n. Remark that similar observations for LTP have been made in [3]. On the other hand, the mechanism of VTRP that increases the transmission range of the particles successfully overcomes these problems. Even for the cases of very low particle densities, VTRP manages to propagate the information reporting the realization of the event to the Sink, with high probability.

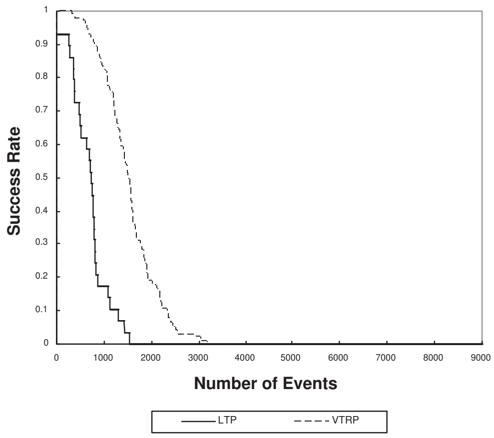


Figure 4: Success Rate ($\mathbb{P}_s$) for LTP and VTRP for multiple events ($n = 5000$).

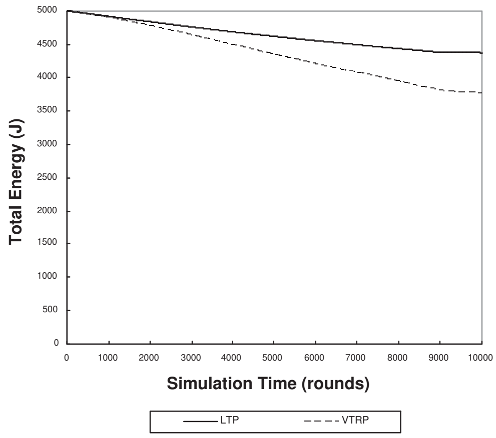


Figure 5: Total Energy (E_{tot}) for LTP and VTRP for multiple events ($n = 5000$).

We continue our experimentation by investigating the performance of the protocols in the case of multiple events. For this set of experiments we drop $n = 5000$ particles uniformly distributed in a 2000m by 2000m square field. Then in each simulation round, we generate one event at a random location in the sensor field that is sensed by only one particle (given that this particle has enough power to sense it), i.e. we use a high event generation rate. This is repeated until a total of 9000 events are generated. Note that this is the first time that the LTP protocol is evaluated under the setting of multiple events.

Figure 4 depicts the success rate of the two protocols as the multiple events are generated. Clearly, VTRP achieves better results than LTP and in fact manages to propagate almost two times more events. The superiority of VTRP is explained by the fact that in LTP the particles that are closer to $\mathcal{S}$ will always participate in the propagation of the messages. The continuous transmissions of messages will eventually exhaust the power of this small group of (highly critical) particles, rendering the rest of the network useless (although there are still energy supplies available) since no further events can be reported to $\mathcal{S}$. VTRP overcomes this problem by activating the *Transmission Range Variation Phase*. As soon as the particles close to $\mathcal{S}$ "die", the neighboring nodes will sense it (during the *Search Phase*) and adjust their transmission range appropriately bypassing them and reaching the sink directly. This is clearly seen in figure 6 where snapshots of the network are taken for different time instances. As soon as some particles around $\mathcal{S}$ "die", LTP fails to deliver the remaining events.

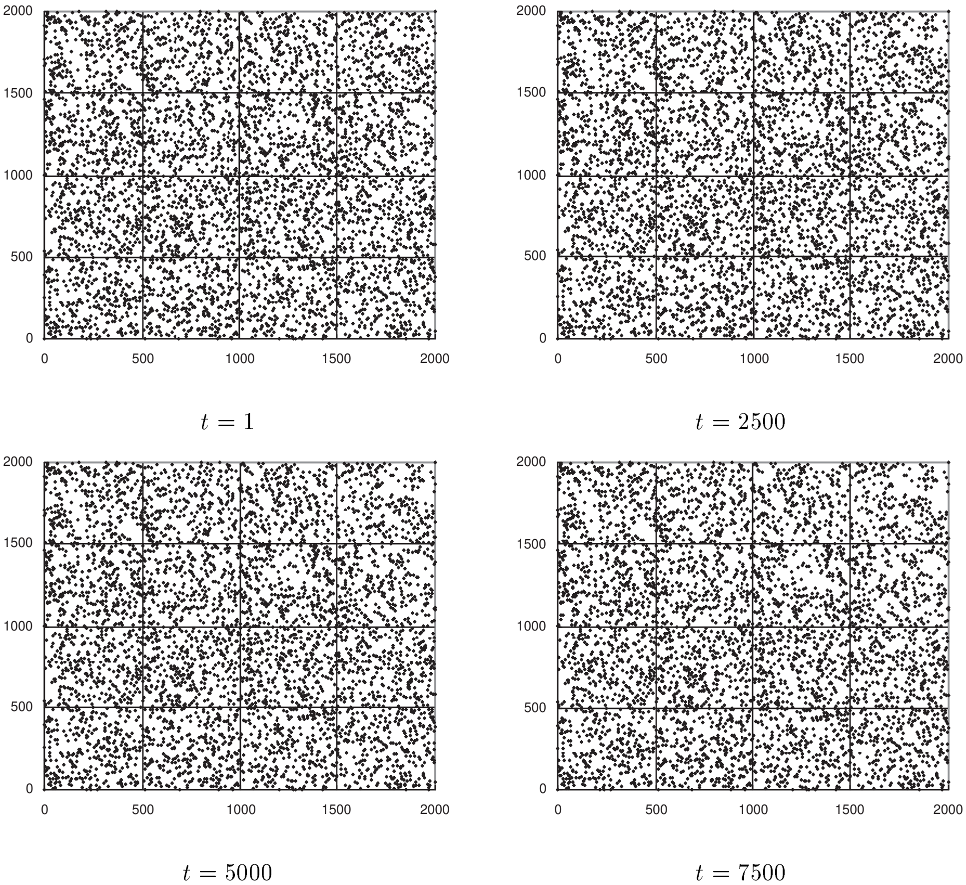
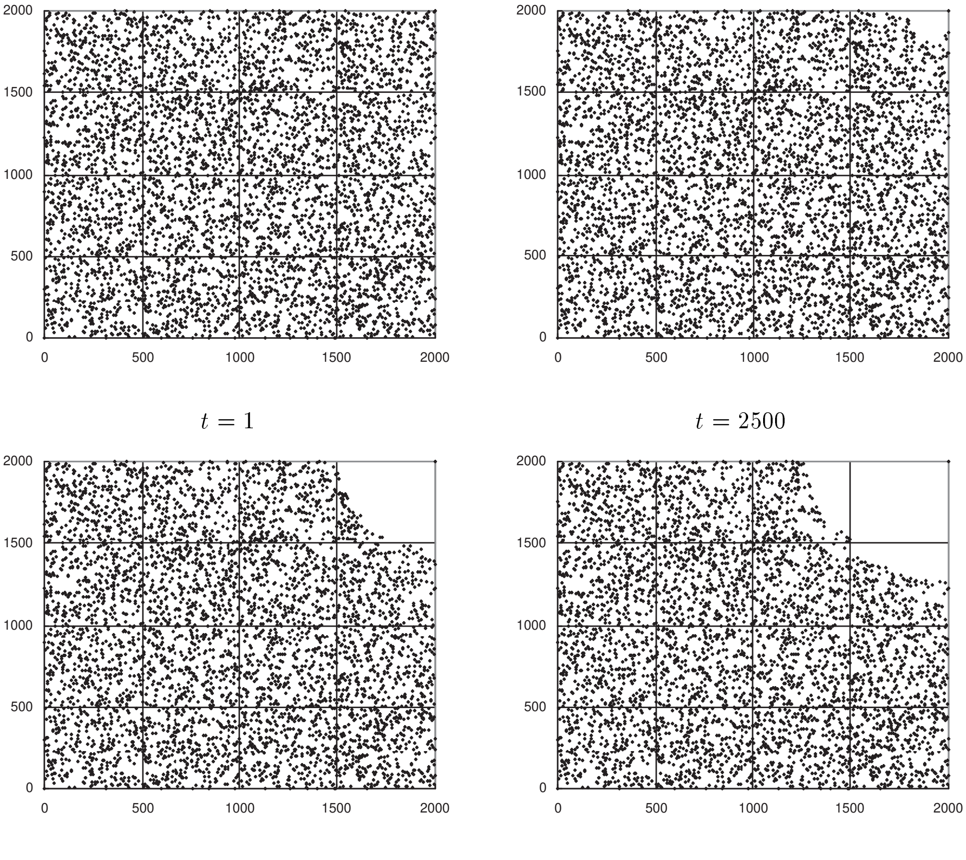


Figure 6: Snapshots of the Network showing alive particles when executing LTP at different time instances ($n = 5000$).

Essentially, VTRP manages the energy of the network in a more efficient way. By examining fig. 5

we observe that VTRP *ends up using slightly more energy than* LTP *in order to propagate more events to the control center*. In fact VTRP will force the particles to spend more energy so that their transmissions manage to reach $\mathcal{S}$ even if this will exhaust their power supplies. Again, this is clearly seen in figure 7 where snapshots of the network are taken for different time instances.



$t = 1$ $t = 2500$

$t = 5000$ $t = 7500$

Figure 7: Snapshots of the Network showing alive particles when executing VTRP at different time instances ($n = 5000$).

To get a more complete view on how each protocol manages the energy resources of the particles, figures 8 and 9 show the number of alive particles based on their distance from the sink. In these figures we have grouped the particles in 32 sets based on the division of the diagonal line connecting $(0, 0)$ with $(2000, 2000)$ in 32 sectors. We observe that for different time instances, the total number of alive particles that are close to the sink (for sections $1 - 10$) drops as the time increases while the particles further away almost always remain active until the end of the experiment. Observe how VTRP forces the particles close to $\mathcal{S}$ to sacrifice their battery supplies in order to propagate more messages.

In the last set of experiments we evaluate the performance of the four different functions for varying the transmission range of the particles when *phase 3* is activated. We use a similar setting as in the previous experiments, i.e. the field size is 2000m by 2000m, we deploy $n = 5000$ sensor and generate 9000 events. The result of this set of experiments are shown in figures 10-15.

The results indicate that the *constant progress* seems to be the least efficient function regarding the *success rate* metric (Fig. 10) while for the other three functions, the achieved success rate seems to be at

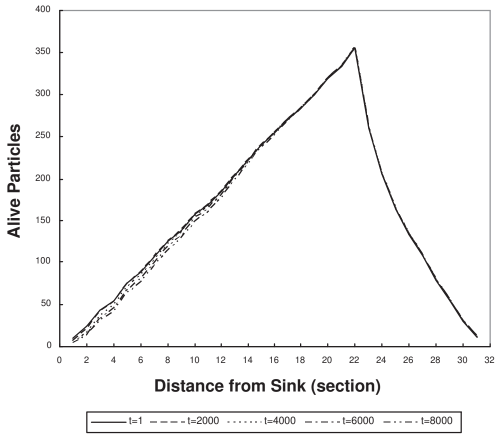


Figure 8: Alive Particles (h_A) for LTP at different time instances ($n = 5000$).

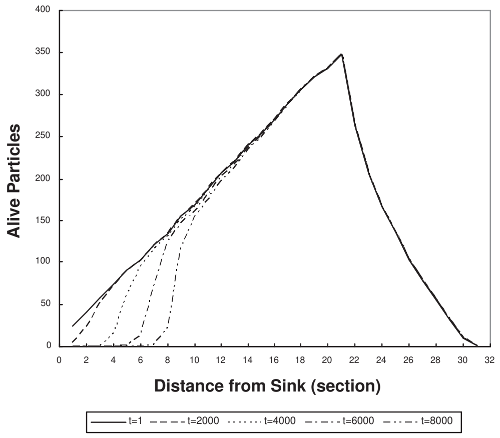


Figure 9: Alive Particles (h_A) for VTRP at different time instances ($n = 5000$).

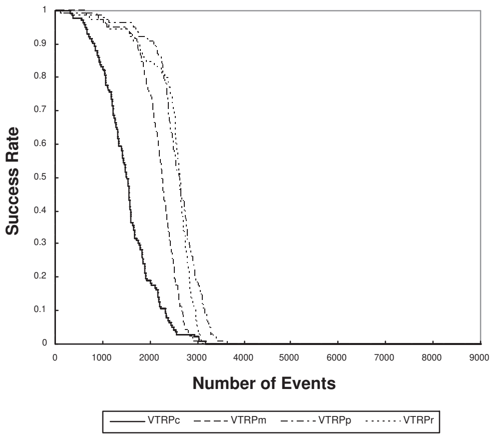


Figure 10: Success Rate ($\mathbb{P}_s$) for the VTRP variations for multiple events ($n = 5000$).

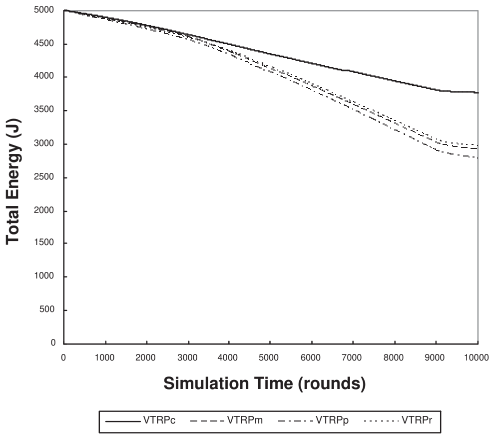


Figure 11: Total Energy (E_{tot}) for the VTRP variations for multiple events ($n = 5000$).

similar levels. In fact this is also the case for the *total energy* consumption (Fig. 11). The *constant progress* function seems to be the most conservative, however, as in the case of LTP, it actually implies that VTRP_C just fails to reach the sink.

A possible explanation to this behavior of VTRP_C is the way the protocol modifies the transmission range by making small, constant steps. At the early stages of the network's operation, when only a small number of particles have "died", these small steps suffice to reach the sink. However, as the distance of the closest still-active particle to $\mathcal{S}$ increases (see fig. 7), the strategy of making small steps becomes inefficient. The series of small increments in the transmission range and failed searches, waste the power sources of the particles and eventually cause the "death" of the particle before the information reaches the sink.

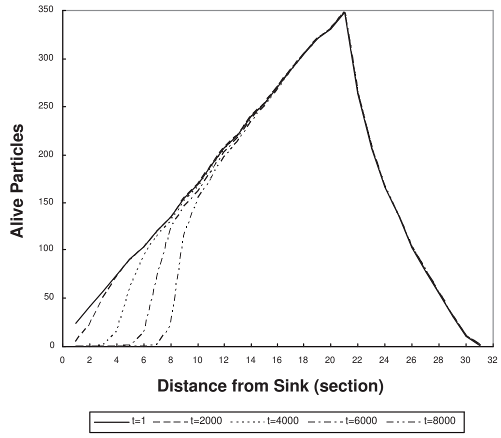


Figure 12: Alive Particles (h_A) for VTRP_c at different time instances ($n = 5000$).

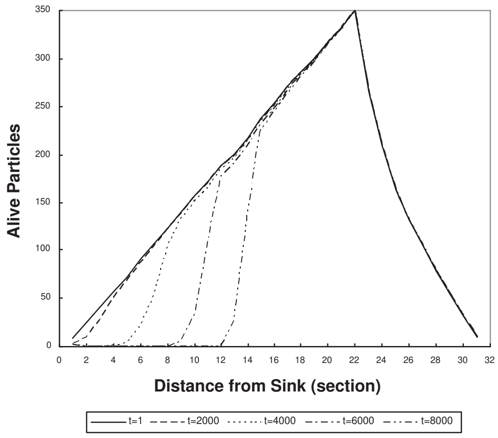


Figure 13: Alive Particles (h_A) for VTRP_m at different time instances ($n = 5000$).

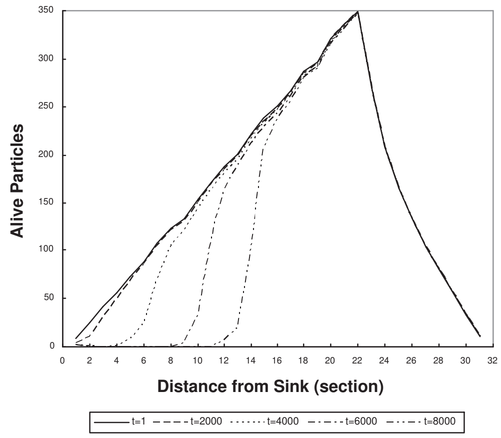


Figure 14: Alive Particles (h_A) for VTRP_p at different time instances ($n = 5000$).

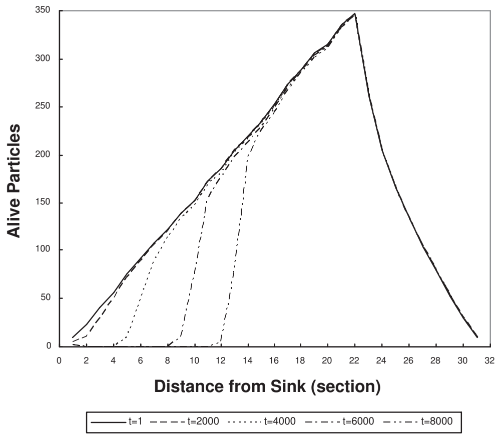


Figure 15: Alive Particles (h_A) for VTRP_r at different time instances ($n = 5000$).

9 Closing Remarks

We presented in this work a new protocol (VTRP) and the extended version of LTP, for multiple event 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 VTRP protocol achieves high success rates regardless of the network density (i.e. even in sparse networks), it performs well in the case of frequent events and operates efficiently in large area networks. On the other hand, the LTP protocol achieves high success rates in networks of high particle densities but there is a deterioration of its performance as the number of events that need to be reported to the control center increases.

We plan to study different network shapes, various distributions used to drop the sensors in the area of interest and the fault-tolerance of the protocols. Finally, we plan to provide performance comparisons with other protocols mentioned in the related work section, as well as investigate different mechanisms for modifying the transmission range and even incorporate an LTP-like backtrack mechanism.

References

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

[2] I. Chatzigiannakis, S. Nikoletseas: A Sleep-Awake Protocol for Information Propagation in Smart Dust Networks. In *Proc. 3rd Workshop on Mobile and Ad-Hoc Networks (WMAN 2002)*, IPDPS Workshops, IEEE Computer Society, Nice, France, Apr. 2003, pp. 225.

[3] I. Chatzigiannakis, S. Nikoletseas and P. Spirakis: Smart Dust Protocols for Local Detection and Propagation. In *Proc. 2nd ACM Workshop on Principles of Mobile Computing (POMC 2002)*, Toulouse, France, Oct. 2002, pp. 9-16.

[4] I. Chatzigiannakis, T. Dimitriou, S. Nikoletseas and P. Spirakis: A Probabilistic Forwarding Protocol for Efficient Data Propagation in Sensor Networks. FLAGS Technical Report, FLAGS-TR-14, 2003.

[5] I. Chatzigiannakis, T. Dimitriou, M Mavronicolas, S. Nikoletseas and P. Spirakis: A Comparative Study of Protocols for Efficient Data Propagation in Smart Dust Networks, Distinguished Paper, in *Proc. 9th International Conference on Parallel and Distributed Computing (EUROPAR 2003)*, Klagenfurt, Austria, Aug. 2003, pp. 1003-1016.

[6] S. Nikoletseas, I. Chatzigiannakis, A. Antoniou, H. Euthimiou, A. Kinalis and G. Mylonas: Energy Efficient Protocols for Sensing Multiple Events in Smart Dust Networks. CTI Technical Report, 2003.

[7] D. Estrin, R. Govindan, J. Heidemann and S. Kumar: Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM 1999)*, Seattle, Washington, USA, Aug. 1999, pp. 263–270.

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

[9] 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)*, Maui, Hawaii, USA, Jan. 2000, pp. 8020.

[10] W. R. Heinzelman, J. Kulik and H. Balakrishnan: Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM 1999)*, Seattle, Washington, USA, Aug. 1999, pp. 174–185.

[11] 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)*.

[12] 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.

[13] J.M. Kahn, R.H. Katz and K.S.J. Pister: Next Century Challenges: Mobile Networking for "Smart Dust". In *Proc. 5th Annual ACM/IEEE International Conference on Mobile Computing (MOBICOM 1999)*, Seattle, Washington, USA, Aug. 1999, pp. 271–278.

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

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

[16] 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 (WPIM 2002)*, IPDPS Workshops, IEEE Computer Society, Ft. Lauderdale, Florida, USA, Apr. 2002, pp. 195b.

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

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

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

[20] W. Ye, J. Heidemann and D. Estrin: An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proc. 21st Annual IEEE International Conference on Computer Communications and Networking (INFOCOM 2002)*, New York, NY, USA, June. 2002, pp. 947–957.

[21] Wireless Integrated Sensor Networks: `http:/www.janet.ucla.edu/WINS/`, April, 2001.