Review

Probabilistic Distributed Algorithms for Energy Efficient Routing and Tracking in Wireless Sensor Networks

Sotiris Nikoletseas ⋆ **and Paul G. Spirakis**

Computer Technology Institute (CTI) and Dept of Computer Engineering and Informatics, University of Patras, Greece; E-mails: nikole@cti.gr; spirakis@cti.gr

⋆ Author to whom correspondence should be addressed

Abstract: In this work we focus on the energy efficiency challenge in wireless sensor networks, from both an on-line perspective (related to routing), as well as a network design perspective (related to tracking). We investigate a few representative, important aspects of energy efficiency: a) the robust and fast data propagation b) the problem of balancing the energy dissipation among all sensors in the network and c) the problem of efficiently tracking moving entities in sensor networks. Our work here is a methodological survey of selected results that have already appeared in the related literature.

In particular, we investigate important issues of energy optimization, like minimizing the total energy dissipation, minimizing the number of transmissions as well as balancing the energy load to prolong the system's lifetime. We review characteristic protocols and techniques in the recent literature, including probabilistic forwarding and local optimization methods.

We study the problem of localizing and tracking multiple moving targets from a network design perspective i.e. towards estimating the least possible number of sensors, their positions and operation characteristics needed to efficiently perform the tracking task. To avoid an expensive massive deployment, we try to take advantage of possible coverage overlaps over space and time, by introducing a novel combinatorial model that captures such overlaps. Under this model, we abstract the tracking network design problem by a covering combinatorial problem and then design and analyze an efficient approximate method for sensor placement and operation.

Keywords: sensor networks; energy efficiency; routing; tracking; distributed algorithms; probabilistic methods.

1. Introduction

1.1. A Brief Description of Wireless Sensor Networks

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 wireless communication capabilities. Current devices have a size at the cubic centimeter scale, a CPU running at 4 MHz, some memory and a wireless communication capability at a 4kbps rate. Also, they are equipped with a small but effective operating system and are able to switch between "sleeping" and "awake" modes to save energy. Pioneering groups (like the "Smart Dust" Project at Berkeley, the "Wireless Integrated Network Sensors" Project at UCLA and the "Ultra low Wireless Sensor" Project at MIT) pursue further important goals, like a total volume of a few cubic millimeters and extremely low energy consumption, by using alternative technologies, based on radio frequency (RF) or optical (laser) transmission.

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.

We note that a single, universal and technology independent model is missing in the state of the art, so in each case we precisely state the particular modeling assumptions (weaker or stronger) needed. Also, in Section 1.3 we make an attempt to define a hierarchy of models and relations between them.

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, 15, 29].

1.2. Critical Challenges

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, analysis and implementation aspects* which are essentially different not only with respect to distributed computing and systems approaches but also to ad-hoc networking techniques.

We emphasize the following characteristic differences between sensor networks and ad-hoc networks:

- The number of sensor particles in a sensor network is extremely large compared to that in a typical ad-hoc network.

- Sensor networks are typically prone to faults.

- Because of faults as well as energy limitations, sensor nodes may (permanently or temporarily) join or leave the network. This leads to highly dynamic network topology changes.

- The density of deployed devices in sensor networks is much higher than in ad-hoc networks.

- The limitations in energy, computational power and memory are much more severe in sensor networks.

Because of the above rather unique characteristics of sensor networks, efficient and robust distributed protocols and algorithms should exhibit the following critical properties:

Scalability

Distributed protocols for sensor networks should be highly scalable, in the sense that they should operate efficiently in extremely large networks composed of huge numbers of nodes. This feature calls for an urgent need to prove by analytical means and also validate (by large scale simulations) certain efficiency and robustness (and their trade-offs) guarantees for asymptotic network sizes.

Efficiency

Because of the severe energy limitations of sensor networks and also because of their time-critical application scenaria, protocols for sensor networks should be efficient, with respect to both energy and time.

Fault-tolerance

Sensor particles are prone to several types of faults and unavailabilities, and may become inoperative (permanently or temporarily). Various reasons for such faults include physical damage during either the deployment or the operation phase, permanent (or temporary) cease of operation in the case of power exhaustion (or energy saving schemes, respectively). The sensor network should be able to continue its proper operation for as long as possible despite the fact that certain nodes in it may fail.

1.3. *Models and Relations between them*

We note that a single, universal, technology- and cost- independent model is missing in the state of the art, so for each protocol and problem we precisely state the particular modeling assumptions (weaker or stronger) needed. Also, we provide the following hierarchy of models and relations between them.

Basic Model M_0
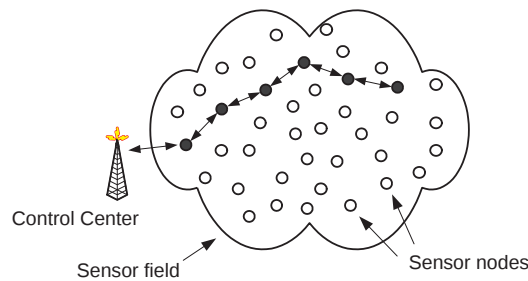
A *smart dust (or sensor) cloud* (a set of sensors, which may be called grain particles) is spread in an area (for a graphical presentation, see Figure 1). The deployment is random uniform. Two important network parameters (crucially affecting connectivity) are the particle density d (usually measured in numbers of $particles/m^2$), and the maximum transmission range $\mathcal{R}$ of each grain particle. The sensors do not move. A two-dimensional framework is adopted.

There is a single point in the network, which is called the "sink" S, that represents the fixed control center where data should be reported to. The sink is very powerful, in terms of both energy and computing power. Sensors can be aware of the direction (and position) of the sink, as well as of their distance to the sink. Such information can be e.g. obtained during a set-up phase, by having the (powerful) sink broadcast control messages to the entire network region.

We assume that events, that should be reported to the control center, occur in the network region. Furthermore, we assume that these events are happening at random uniform positions.

As far as energy dissipation is concerned, we assume that the energy spent at a sensor when transmitting data is proportional to the square of the transmitting distance. Note however that the energy dissipation for receiving is not always negligible with respect to the energy when sending such as in case when transmissions are too short and/or radio electronics energy is high [23].

Figure 1. A Smart Dust Cloud.

A variation of this basic model includes multiple and/or mobile sinks. The following wall notion generalizes that of the sink and may correspond to multiple (and/or moving) sinks. A *receiving wall* $\mathcal{W}$ is defined to be an infinite line in the smart-dust plane. Any particle transmission within range $\mathcal{R}$ from the wall $\mathcal{W}$ is received by $\mathcal{W}$. $\mathcal{W}$ is assumed to have very strong computing power, is able to collect and analyze received data and has a constant power supply and so it has no energy constraints. The wall represents in fact the authorities (the fixed control center) which the realization of a crucial event and collected data should be reported to. Note that a wall of appropriately big (finite) length may suffice. The notion of multiple sinks which may be static or moving has also been studied in [46], where Triantafilloy, Ntarmos, Nikoletseas and Spirakis introduce "NanoPeer Worlds", merging notions from Peer-to-Peer Computing and Smart Dust.

The M_0 model is assumed for the LTP protocol.

A stronger model M_1

This model is derived by augmenting model M_0 with additional (stronger) assumptions about the abilities of the sensors and the sensor deployment. In particular, the network topology can be a lattice (or grid) deployment of sensors. This grid placement of grain particles is motivated by certain applications, where it is possible to have a pre-deployed sensor network, where sensors are put (possibly by a human or a robot) in a way that they form a *2-dimensional lattice*. Note indeed that such sensor networks, deployed in a structured way, might be useful e.g. in precise agriculture, where humans or robots may want to deploy the sensors in a lattice structure to monitor in a rather homogenous and uniform way certain conditions in the spatial area of interest. Also, exact terrain monitoring in military applications may also need some sort of a grid-like shaped sensor network. Note that Akyildiz *et al.* in a state of the art survey refer to the pre-deployment possibility [1]. Also, [25] explicitly refers to the lattice case. Moreover, as the authors of [25] state in an extended version of their work, they consider, for reasons of "analytic tractability", a square grid topology [26].

We also assume that each grain particle has the following abilities:

(i) It can estimate the direction of a received transmission (e.g. via the technology of direction-sensing antennae). We note that at least a gross sense of direction may be technologically possible by smart and/or multiple antennas. Regarding the extra cost of the antennas and circuitry, we note that certainly such a cost is introduced but as technology advances this cost may lower.

(ii) It can estimate the distance from a nearby particle that did the transmission (e.g. via estimation of the attenuation of the received signal).

(iii) It knows the direction towards the sink S. This can be implemented during a set-up phase, where the (very powerful in energy) sink broadcasts the information about itself to all particles.

(iv) All particles have a common co-ordinates system.

Notice that GPS information is assumed by this model. Also, the global structure of the network is not assumed known.

This model is used in the PFR protocol.

Variations

Both models above can be augmented by assuming certain additional abilities, like the ability of sensors to vary their wireless transmission range (this is used in the EBP and the tracking protocols). Also, problem specific assumptions may apply (such as the model for the targets in the tracking problem).

1.4. The Energy Efficiency Challenge in Routing and Tracking

Since one of the most severe limitations of sensor devices is their limited energy supply, one of the most crucial goals in designing efficient protocols for wireless sensor networks is minimizing the energy consumption in the network. This goal has various aspects, including: (a) minimizing the total energy spent in the network (b) minimizing the number (or the range) of data transmissions (c) combining energy efficiency and fault-tolerance, by allowing redundant data transmissions which however should be optimized to not spend too much energy (d) maximizing the number of "alive" particles over time, thus prolonging the system's lifetime and (e) balancing the energy dissipation among the sensors in the network, in order to avoid the early depletion of certain sensors and thus the breakdown of the network.

We note that it is very difficult to achieve all the above goals at the same time. There even exists trade-offs between some of the goals above. Furthermore, the importance and priority of each of these goals may depend on the particular application. Thus, it is important to have a variety of protocols, each of which may possibly focus at some of the energy efficiency goals above (while still performing well with respect to the rest goals as well).

In this paper, we focus on energy optimization. We investigate both on-line aspects of energy efficiency related to data propagation, as well as off-line network design issues, related to tracking, an important problem which is considered canonical in the literature. In particularly for routing we present three methodologically characteristic energy efficient protocols:

- *The Local Target Protocol (LTP)*, that performs a local optimization trying to minimize the number of data transmissions.

- *The Probabilistic Forwarding Protocol (PFR)*, that creates redundant data transmissions that are probabilistically optimized, to trade-off energy efficiency with fault-tolerance.

- *The Energy Balanced Protocol (EBP)*, that focuses on guaranteeing the same per sensor energy dissipation, in order to prolong the lifetime of the network.

Because of the complex nature of a sensor network (that integrates various aspects of communication and computing), protocols, algorithmic solutions and design schemes for all layers of the networking infrastructure are needed. Far from being exhaustive, we mention the need for frequency management solutions at the physical layer, for Medium Access Control (MAC) protocols to cope with multi-hop transmissions at the data link layer. Our approach focuses on routing and does not consider other important issues like link fading and hidden station interference, i.e. we assume that other network layers are resolving these issues. Also, regarding reliability, an endemic issue to wireless sensor networks, we consider permanent physical failures i.e. we do not deal with bad values from physical sensing or transient failures. The interested reader may use the excellent survey by Akyildiz *et al.* [4] for a detailed discussion of design aspects of all layers of the networking infrastructure.

For tracking, we study the problem of localizing and monitoring multiple moving targets in wireless sensor networks, from a network design perspective i.e. towards estimating the least possible number of sensors to be deployed, their positions and operation characteristics needed to efficiently perform the tracking task. To avoid an expensive massive deployment, we try to take advantage of possible coverage overlaps over space and time, by introducing a novel combinatorial model that captures such overlaps.

Under this model, we abstract the tracking network design problem by a combinatorial problem of covering a universe of elements by at least three sets to ensure that each point in the network area is covered at any time by at least three sensors, and thus being localized, i.e. we assume that 3-coverage suffices for (at least gross) localization; our method can be extended to k-coverage, for constant $k \geq 4$, for more accuracy. We then design and analyze an efficient approximate method for sensor placement and operation, that with high probability and in polynomial expected time achieves a $\Theta(\log n)$ approximation ratio to the optimal solution. Our network design solution can be combined with alternative collaborative processing methods, to suitably fit different tracking scenaria.

Part A: On-Line Energy Optimization

2. LTP: A Hop-by-Hop Data Propagation Protocol

The LTP Protocol was introduced in [8]. The authors assume the basic model M_0 defined in Section 1.3.

2.1. The Protocol

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

- *Search Phase*: It uses a periodic low energy directional broadcast in order to discover a particle nearer to $\mathcal{W}$ than itself. (i.e. a particle p'' where $d(p'', \mathcal{W}) < d(p', \mathcal{W})$).

- *Direct Transmission Phase*: Then, p' sends *info(ε)* to p''.

- *Backtrack Phase*: If consecutive repetitions of the *search phase* fail to discover a particle nearer to $\mathcal{W}$, then p' sends *info(ε)* to the particle that it originally received the information from.

Note that one can estimate an a-priori upper bound on the number of repetitions of the search phase needed, by calculating the probability of success of each search phase, as a function of various parameters (such as density, search angle, transmission range). This bound can be used to decide when to backtrack.

For a graphical representation see Figures 2 and 3.

Figure 2. Example of the Search Phase.
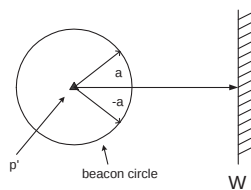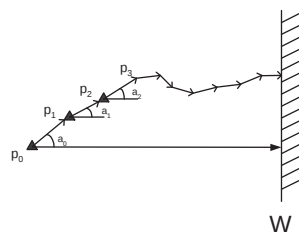
Figure 3. Example of a Transmission.

2.2. Theoretical Analysis

[8] first provides some basic definitions.

Definition 1 Let $h_{opt}(p, \mathcal{W})$ be the (optimal) number of "hops" (direct, vertical to $\mathcal{W}$ transmissions) needed to reach the wall, in the *ideal* case in which particles always exist in pair-wise distances $\mathcal{R}$ on the vertical line from p to $\mathcal{W}$. Let Π be *a smart-dust propagation protocol*, using *a transmission path* of length $L(\Pi, p, \mathcal{W})$ to send info about event $\mathcal{E}$ to wall $\mathcal{W}$. Let $h(\Pi, p, \mathcal{W})$ be the actual number of hops (transmissions) taken to reach $\mathcal{W}$. The *"hops" efficiency* of protocol Π is the ratio

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

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

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

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

Note that any given distribution of particles in the smart dust cloud may not allow the ideal optimal number of hops to be achieved at all. In fact, the least possible number of hops depends on the input (the positions of the grain particles). [8] however, compares the efficiency of protocols to the ideal case. A comparison with the best achievable number of hops in each input case will of course give better efficiency ratios for protocols.

To enable a first step towards a rigorous analysis of smart dust protocols, [8] makes the following simplifying assumption: *The search phase always finds a p'' (of sufficiently high battery) in the semicircle* of center the particle p' currently possessing the information about the event and radius R, in the direction towards $\mathcal{W}$. Note that this assumption on always finding a particle can be relaxed in the following ways: (a) by repetitions of the search phase until a particle is found. This makes sense if at least one particle exists but was sleeping during the failed searches, (b) by considering, instead of just the semicircle, a cyclic sector defined by circles of radiuses $\mathcal{R} - \Delta\mathcal{R}$, $\mathcal{R}$ and also take into account the density of the smart cloud, (c) if the protocol during a search phase ultimately fails to find a particle towards the wall, it may *backtrack*.

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

The above assumptions may not be very realistic in practice, however, they can be relaxed and in any case allow to perform a first effort towards providing some concrete analytical results.

Lemma 1 ([8]) *The expected "hops efficiency" of the local target protocol in the a-uniform case is*

$$E(C_h) \simeq \frac{\alpha}{\sin \alpha}$$

for large h_{opt}. Also

$$1 \leq E(C_h) \leq \frac{\pi}{2} \simeq 1.57$$

for $0 \leq \alpha \leq \frac{\pi}{2}$.

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

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

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

$$\sum_{i=1}^{h-1} \cos \alpha_i \leq h_{opt} \leq \sum_{i=1}^{h} \cos \alpha_i$$

From Wald's equation for the expectation of a sum of a random number of independent random variables [42], then

$$E(h-1) \cdot E(\cos \alpha_i) \leq E(h_{opt}) = h_{opt} \leq E(h) \cdot E(\cos \alpha_i)$$

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

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

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

$$\diamond$$

2.3. *Local Optimization: The Min-two Uniform Targets Protocol (M2TP)*

[8] further assumes that the search phase always returns *two points* p'', p''' each uniform in $(-\alpha, \alpha)$ and that a modified protocol M2TP selects the best of the two points, with respect to the local (vertical) progress. This is in fact an optimized version of the Local Target Protocol.

In a similar way as in the proof of the previous lemma, the authors prove the following result:

Lemma 2 ([8]) *The expected "hops efficiency" of the "min two uniform targets" protocol in the a-uniform case is*

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

for $0 \leq \alpha \leq \frac{\pi}{2}$ and for large h.

Now remark that

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

and

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

Thus, [8] proves the following:

Lemma 3 ([8]) *The expected "hops" efficiency of the min-two uniform targets protocol is*

$$1 \le E(C_h) \le \frac{\pi^2}{8} \simeq 1.24$$

for large h and for $0 \le \alpha \le \frac{\pi}{2}$.

Remark that, with respect to the expected hops efficiency of the local target protocol, the min-two uniform targets protocol achieves, because of the one additional search, a relative gain which is $(\pi/2 - \pi^2/8)/(\pi/2) \simeq 21.5\%$. [8] also experimentally investigates the further gain of additional (i.e. $m > 2$) searches.

3. PFR - A Probabilistic Forwarding Protocol

The LTP protocol, as shown in the previous section manages to be very efficient by always selecting exactly one next-hop particle, with respect to some optimization criterion. Thus, it tries to minimize the number of data transmissions. LTP is indeed very successful in the case of dense and robust networks, since in such networks a next hop particle is very likely to be discovered. In sparse or faulty networks however, the LTP protocol may behave poorly, because of many backtracks due to frequent failure to find a next hop particle. To combine energy efficiency and fault-tolerance, the Probabilistic Forwarding Protocol (PFR) has been introduced. The trade-offs in the performance of the two protocols implied above are shown and discussed in great detail in [10].

The PFR protocol was introduced in [9]. Authors assume the model M_1 (defined in section 1.3).

3.1. The Protocol

The PFR protocol is inspired by the probabilistic multi-path design choice for the Directed Diffusion paradigm mentioned in [25]. Its basic idea of the protocol (introduced in [9]) lies in probabilistically favoring 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 Figure 4). Note that transmission along this line is energy optimal. However it is not always possible to achieve this optimality, basically because certain sensors on this direct line might be inactive, either permanently (because their energy has been exhausted) or temporarily (because these sensors might enter a sleeping mode to save energy). Further reasons include:

- physical damage of sensors,

- deliberate removal of some of them (possibly by an adversary in military applications),

- changes in the position of the sensors due to a variety of reasons (weather conditions, human interaction *etc.*), and

- physical obstacles blocking communication.

The protocol evolves in two phases:

Phase 1: The "Front" Creation Phase. Initially the protocol builds (by using a limited, in terms of rounds, flooding) a sufficiently large "front" of particles, in order to guarantee the survivability of the data propagation process. During this phase, each particle having received the data to be propagated, deterministically forwards them towards the sink. In particular, and for a sufficiently large number of steps $s = 180\sqrt{2}$, each particle broadcasts the information to all its neighbors, towards the sink. Remark that to implement this phase, and in particular to count the number of steps, we use a counter in each message. This counter needs at most $\lceil \log 180\sqrt{2} \rceil$ bits.

Phase 2: The Probabilistic Forwarding Phase. During this phase, each particle P possessing the information under propagation, calculates an angle ϕ by calling the subprotocol "ϕ-calculation" (see description below) and broadcasts $info(\mathcal{E})$ to all its neighbors with probability $\mathbb{P}_{fwd}$ (or it does not propagate any data with probability $1 - \mathbb{P}_{fwd}$) defined as follows:

$$\mathbb{P}_{fwd} = \begin{cases} 1 & \text{if } \phi \geq \phi_{threshold} \\ \frac{\phi}{\pi} & \text{otherwise} \end{cases}$$

where ϕ is the angle defined by the line EP and the line PS and $\phi_{threshold} = 134^o$ (the selection reasons of this $\phi_{threshold}$ can be found in [9]).

In both phases, if a particle has already broadcast $info(\mathcal{E})$ and receives it again, it ignores it. Also the PFR protocol is presented for a single event tracing. Thus no multiple paths arise and packet sizes do not increase with time.

Remark that when $\phi = \pi$ then P lies on the line ES and vice-versa (and always transmits).

If the density of particles is appropriately large, then for a line ES there is (with high probability) a sequence of points "closely surrounding ES" whose angles ϕ are larger than $\phi_{threshold}$ and so that successive points are within transmission range. All such points broadcast and thus essentially they follow the line ES (see Figure 4).

**Figure 4.** Thin Zone of particles.
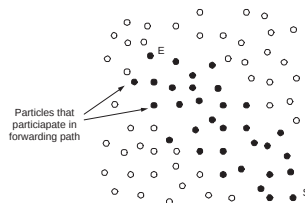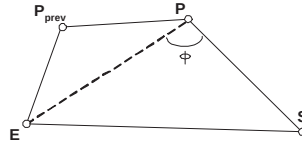
Figure 5. Angle ϕ calculation example.

The ϕ-calculation subprotocol (see Figure 5)

Let P_{prev} the particle that transmitted $info(E)$ to P.

(1) When P_{prev} broadcasts $info(E)$, it also attaches the info $|EP_{prev}|$ and the direction $\overrightarrow{P_{prev}E}$.

(2) P estimates the direction and length of line segment $P_{prev}P$, as described in the model.

(3) P now computes angle $(\widehat{EP_{prev}P})$, and computes $|EP|$ and the direction of $\overrightarrow{PE}$ (this will be used in further transmission from P).

(4) P also computes angle $(\widehat{P_{prev}PE})$ and by subtracting it from $(\widehat{P_{prev}PS})$ it finds ϕ.

Notice the following:

(i) The direction and distance from activated sensors to E is inductively propagated (i.e. P becomes P_{prev} in the next phase).

(ii) The protocol needs only messages of length bounded by $\log A$, where A is some measure of the size of the network area, since (because of (i) above) there is no cumulative effect on message lengths.

Essentially, the protocol captures the intuitive, deterministic idea "if my distance from ES is small, then send, else do not send". [9] has chosen to enhance this idea by random decisions (above a threshold) to allow some local flooding to happen with small probability and thus to cope with local sensor failures.

3.2. Properties of PFR
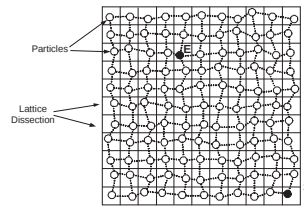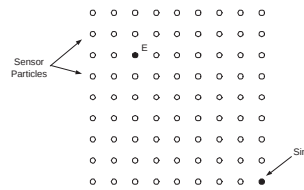
Any protocol Π solving the data propagation problem must satisfy the following three properties:

- **Correctness**. Π must guarantee that data arrives to the position S, given that the whole network exists and is operational.

- **Robustness**. Π must guarantee that data arrives at enough points in a small interval around S, in cases where part of the network has become inoperative.

- **Efficiency**. If Π activates k particles during its operation then Π should have a small ratio of the number of activated over the total number of particles $r = \frac{k}{N}$. Thus r is an energy efficiency measure of Π.

[9] shows that this is indeed the case for PFR.

Consider a partition of the network area into small squares of a fictitious grid G (see Figure 6). Let the length of the side of each square be l. Let the number of squares be q. The area covered is bounded by ql^2. Assuming that we randomly throw in the area at least $\alpha q \log q = N$ particles (where $\alpha > 0$ a suitable constant), then the probability that a particular square is avoided tends to 0. So with very high probability (tending to 1) all squares get particles.

[9] conditions all the analysis on this event, call it F, of at least one particle in each square.

Figure 6. A Lattice Dissection G.

Figure 7. A Lattice Sensor Network.

3.3. The Correctness of PFR

Without loss of generality, we assume each square of the fictitious lattice G to have side length 1.

In [9] the authors prove the correctness of the PFR protocol, by using a geometric analysis. We below sketch their proof.

Consider any square Σ intersecting the ES line. By the occupancy argument above, there is with high probability a particle in this square. Clearly, the worst case is when the particle is located in one of the corners of Σ (since the two corners located most far away from the ES line have the smallest ϕ-angle among all positions in Σ).

By some geometric calculations, [9] finally proves that the angle ϕ of this particle is $\phi > 134^o$. But the initial square (i.e. that containing E) always broadcasts and any intermediate intersecting square will be notified (by induction) and thus broadcast because of the argument above. Thus the sink will be reached if the whole network is operational.

Lemma 4 ([9]) PFR *succeeds with probability 1* in sending the information from E to S given the event F.

3.4. The Energy Efficiency of PFR

[9] considers the fictitious lattice G of the network area and let the event F hold. There is (at least) one particle inside each square. Now join all nearby particles of each particle to it, thus by forming a new graph G' which is "lattice-shaped" but its elementary "boxes" may not be orthogonal and may have varied length. When squares of G become smaller and smaller, then G' will look like G. Thus, for reasons of analytic tractability, in [9] the authors assume that particles form a lattice (see Figure 7). They also assume length $l = 1$ in each square, for normalization purposes. Notice however that when $l \to 0$ then "$G' \to G$" and thus all results in this Section hold for any random deployment "in the limit".

The analysis of the energy efficiency considers particles that are active but are as far as possible from ES. Thus the approximation is suitable for remote particles.

[9] estimates an upper bound on the number of particles in an $n \times n$ (i.e. $N = n \times n$) lattice. If k is this number then $r = \frac{k}{n^2}$ ($0 < r \leq 1$) is the "energy efficiency ratio" of PFR.

More specifically, in [9] the authors prove the (very satisfactory) result below. They consider the area around the ES line, whose particles participate in the propagation process. The number of active particles is thus, roughly speaking, captured by the size of this area, which in turn is equal to $|ES|$ times the maximum distance from $|ES|$ (where maximum is over all active particles).

This maximum distance is clearly a random variable. To calculate the expectation and variance of this variable, the authors in [9] basically "upper bound" the stochastic process of the distance from ES by a random walk on the line, and subsequently "upper bound" this random walk by a well-known stochastic process (i.e. the "discouraged arrivals" birth and death Markovian process, see e.g. [31]). Thus they prove the following:

Theorem 1 ([9]) *The energy efficiency of the* PFR *protocol is* $\Theta\left(\left(\frac{n_0}{n}\right)^2 \right)$ *where* $n_0 = |ES|$ *and* $n = \sqrt{N}$, *where* N *is the number of particles in the network. For* $n_0 = |ES| = o(n)$, *this is* $o(1)$.

3.5. The Robustness of PFR

To prove the following robustness result, the authors in [9] consider particles "very near" to the ES line. Clearly, such particles have large ϕ-angles (i.e. $\phi > 134^o$). Thus, even in the case that some of these particles are not operating, the probability that none of those operating transmits (during the probabilistic phase 2) is very small. Thus, [9] proves the following.
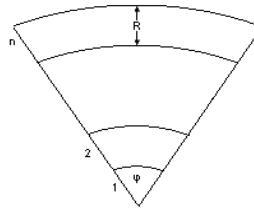
Lemma 5 ([9]) PFR *manages to propagate the crucial data across lines parallel to* ES, *and of constant distance, with* fixed *nonzero probability (not depending on* n, $|ES|$*).*

4. The Energy Balance Problem

In order to save energy and keep the network functional for as long as possible, various approaches, including hop-by-hop transmission techniques [8, 25, 26], as well as clustering techniques [23] and alternating power-saving modes [44] have been proposed.

All such techniques, in their basic form presented in sections 2 and 3, do not *explicitly* take care of the possible overuse of certain sensors in the network. As an example, remark that in hop-by-hop transmissions towards the sink, the sensors lying closer to the sink tend to be utilized exhaustively (since all data passes through them). Thus, these sensors may die out very early, thus resulting to network collapse although there may be still significant amounts of energy in the other sensors of the network. Similarly, in clustering techniques the cluster-heads that are located far away with respect to the sink, tend to spend a lot of energy.

In this paper, we present the EBP (Energy Balance) protocol, introduced in [16], that probabilistically chooses between either propagating data one hop towards the sink or sending directly to the sink. The first choice is more energy efficient, while the latter bypasses the critical (close to the sink) sectors. The appropriate probability for each choice in order to achieve energy balance is calculated in [16]. We also mention a related, extended protocol (VTRP: Variable Transmission Range Protocol), proposed in [3], that implicitly contributes to energy balance by appropriately adapting (increasing) the transmission range, thus bypassing critical sensors and avoiding possible obstacles.

Figure 8. Sensor Network with n ring sectors, angle ϕ and ring "width" R

4.1. The Model and the Problem

The basic model M_0 (see section 1.3) is assumed. Additionally, we assume that the transmission range of sensors can vary with time (in fact, for each sensor our protocol may use only two different ranges: R and $i \cdot R$, where i is a measure of the sensor's distance to the sink).

We virtually "cover" the network area by a cycle sector of angle ϕ (see Figure 8). Similar ring formation by flooding of beacon messages with hop count has been proposed in [54]. The cycle sector is divided into n ring sectors or "*slices*". The first slice has radius R (i.e. the sensors' transmission range). Slice i ($2 \leq i \leq n$) is defined by two cycles sectors, one of radius $i \cdot R$ and the other of radius $(i-1) \cdot R$. We do not address what is the largest allowable transmission range (e.g. the largest i above); this would be limited by technology and interferences. However, taking a sufficiently large angle ϕ and/or by hierarchically taking multiple sectors, we can cover the whole area without increasing i beyond the maximum allowable limit.

In the analysis, we only count (for simplicity) energy spent during transmissions. Since however in our protocol (see next section) there is one receipt for each transmission, it is clear that even when energy during receipt is more or less the same as energy during transmissions, the analysis can be extended easily to the full case (counting both transmissions and receipts). We do not consider energy aspects due to duty-cycling, i.e. we assume that an appropriate power saving scheme is resolving this.

Definition 2 The area between two consecutive cycle sectors is called *a ring sector* (or "slice"). Let T_i ($1 \leq i \leq n$) be the i-th ring sector of the network.

T_1 stands for the ring sector with center the sink and radius equal to R.

Definition 3 Let S_i be the area size of the ring sector T_i of the network ($1 \leq i \leq n$).

We wish to solve the "*energy balanced data propagation problem*", i.e. to propagate data to the sink in such a way that the "average" energy dissipation in each sensor is at each time the same. The average energy dissipation per sensor is taken to be the fraction of the total energy spent by sensors in a ring sector over the number of sensors in that sector. Because of our assumption that the number of sensors in an area is proportional to the area size, the average energy dissipation per sensor is calculated by dividing the total energy spent in a sector by the sector size.

We do not study here medium access aspects, assuming the existence of underlying MAC protocol.

4.2. EBP: The Energy Balance Protocol

We assume that each event is sensed by only one sensor. This assumption is not restrictive since we may consider multiple sensing and propagation of an event by various sensors as sensing and propagation of many different events, assuming methods to resolve symmetry (such as clustering, leader election) in cases like when sensors are acoustic and the event is a thunderclap. A sensor sensing an event generates then a data message which should be eventually delivered to the sink. On each ring sector, T_i, a number of events occur and a corresponding number of messages (one for each event) is generated.

Randomization is used to achieve some "load balancing" by evenly spreading the "load" (energy dissipation). In particular, on ring sector T_i each event is propagated to T_{i-1} (i.e. the "*next*" sector towards the sink) with probability p_i, while with probability $1 - p_i$ it is propagated directly to the sink S. Each message in T_i is handled stochastically independently of the other events' messages.

The choice of probability p_i for T_i is made so as the average energy consumption per area unit (and thus per sensor) is the same for the whole network. There is a trade-off from choosing p_i: if p_i increases then transmissions tend to happen locally, thus energy consumption is low, however sensors closer to the sink tend to be overused since all data passes through them. On the other hand, if p_i decreases, there are distant transmissions (thus a lot of energy is consumed) however closer to sink particles are bypassed. Calculating the appropriate probability p_i for each T_i and solving the problem of energy balance is very important since it combines efficient data propagation with increased network's lifetime.

By using an underlying subprotocol [8, 25] we can guarantee that only one "next hop" sensor receives the transmitted message. Note also that data messages are of fixed size i.e. no further info is added to a message on its route towards the sink.

Our protocol is (a) *distributed*, since each sensor chooses propagation probability independently of other sensors, (b) it uses only *local* information, in the sense that p_i depends only on i, i.e. a parameter related to the distance from the sink. Note that distance from the sink info for each sensor can be easily obtained i.e. during a set-up phase where the sink broadcasts control messages to the network. Several techniques (including signal attenuation evaluation) can be used to estimate each sensor's distance for the sink. (c) The protocol is very *simple*, since it just uses a random choice based only on parameter i.

4.3. Basic Definitions - Preliminaries

We aim at calculating probability p_i for each i in order to ensure the energy balance property. Using simple geometry, one can easily prove the following Lemmas.

Lemma 6 The area size, S_1, of the ring sector T_1 is $S_1 = \frac{\phi}{2} \cdot R^2$

Lemma 7 The relation between the area size of the ring sector T_i and that of T_1 is $S_i = (2i - 1) \cdot S_1$

Definition 4 Let λ_i the probability that an event will occur on the ring sector T_i.

There are n ring sectors in the network.

Lemma 8 Assuming a random uniform generation of events in the network area, the probability λ_i of an event occurring on the ring sector T_i $(1 \leq i \leq n)$, is:

$$\lambda_i = \frac{(2i - 1)}{n^2}$$

Let us now consider sector T_i.

Definition 5 An area T_i *"handles"* an event generated in ring sector j if either the message was generated in the area T_i (i.e. $j = i$) or the message was propagated to T_i from the ring sector T_{i+1}.

Definition 6 Let h_i be the number of the messages that are "handled" by the area T_i.

We now define energy ϵ_{ij} spent for message j when sector i handles it.

Definition 7 Let ϵ_{ij} a random variable which measures the energy that dissipates the sector T_i so as to handle the message j. For ϵ_{ij} we have that:

$$\epsilon_{ij} = \begin{cases} cR^2 & \text{with probability } p_i \\ c(iR)^2 & \text{with probability } 1 - p_i \end{cases}$$

where cR^2 is the energy dissipation for sending a message j from T_i to its adjacent ring sector T_{i-1} and c is a constant.

Thus, the expected energy dissipation in sector i for handling a message is

$$E[\epsilon_{i,j}] = cR^2 \cdot [i^2 - p_i(i^2 - 1)] \tag{1}$$

Note: The expected energy above is the same for all messages; we use j just for counting purposes.

Definition 8 Let $\mathcal{E}_i$ the *total energy* spent by sensors in T_i. Clearly:

$$\mathcal{E}_i = \sum_{j=1}^{h_i} \epsilon_{ij} \tag{2}$$

Energy balance is defined as follows:

Definition 9 The network is energy balanced if the average per sensor energy dissipation is the same for all sectors, i.e. when

$$\frac{E[\mathcal{E}_i]}{S_i} = \frac{E[\mathcal{E}_j]}{S_j} \quad i, j = 1, \ldots, n \tag{3}$$

4.4. The General Solution

We next provide a lemma useful in the estimation of the total energy dissipation in a sector.

Lemma 9 The expected total energy dissipation in sector i is:

$$E[\mathcal{E}_i] = E[h_i] \cdot E[\epsilon_{ik}]$$

Proof:

$$\begin{aligned} E[\mathcal{E}_i] &= E\left[\sum_{k=1}^{h_i} \epsilon_{ik}\right] \\ &= \sum_{n=0}^{N} E\left[\sum_{k=0}^{h_i} (\epsilon_{ik} \cap h_i = n)\right] \\ &= \sum_{n=0}^{N} E\left[\sum_{k=1}^{h_i} \epsilon_{ik} \Big| h_i = n\right] \cdot \mathbb{P}\{h_i = n\} \end{aligned}$$

Furthermore,

$$E\left[\sum_{k=1}^{h_i} \epsilon_{ik} \Big| h_i = n\right] = E\left[\sum_{k=1}^{n} \epsilon_{ik}\right] = \sum_{k=1}^{n} E[\epsilon_{ik}]$$
$$= n \cdot E[\epsilon_{ik}]$$

Thus, we have from the above that:

$$E[\mathcal{E}_i] = \sum_{n=0}^{N} E\left[\sum_{k=1}^{h_i} \epsilon_{ik} \Big| h_i = n\right] \cdot \mathbb{P}\{h_i = n\}$$
$$= E[\epsilon_{ik}] \cdot E[h_i]$$

◇

Definition 10 Let g_i be the number of the messages that are *generated* in the area T_i.

Note that messages are generated in an area only when events occur in this area.

Definition 11 Let f_i be the number of the messages that are *forwarded to* the area T_i.

We note that messages are forwarded to a ring sector (say i) only because of an event generation at a sector $j > i$ and successive one-hop propagations from sector j to sector i.
We notice the following important relation:

$$h_i = g_i + f_i \tag{4}$$

which means that the number of messages that area T_i handles equals the number of the messages that are generated in T_i, plus the number of messages that are forwarded to it. Because of event generation according to a probability distribution and also because of the probabilistic nature of message propagation in, all three quantities above are random variables. By linearity of expectation, we get:

Lemma 10 $E[h_i] = E[g_i] + E[f_i]$

We establish a relationship between $E[f_i]$ and $E[h_{i+1}]$.

Lemma 11 $E[f_i] = p_{i+1} \cdot E[h_{i+1}]$

Proof: Let $\delta_{i,j}$ an indicator random variable that is equal to 1 if area T_i forwards the message j to the area T_{i-1} and 0 otherwise. Thus:

$$\delta_{i,j} = \begin{cases} 1 & \text{with probability } p_i \\ 0 & \text{with probability } 1 - p_i \end{cases}$$

Clearly, $\delta_{i,j}$ depends only on i, but we add j for counting purposes. Obviously, $E[\delta_{i,j}] = p_i$. It is:

$$f_i = \sum_{j=0}^{h_{i+1}} \delta_{i+1,j}$$

Similarly to the proof of Lemma 9, we get:

$$E[f_i] = \sum_{n=0}^{N} E\left[\sum_{j=0}^{h_{i+1}} \delta_{i+1,j} \Big| h_{i+1} = n\right] \cdot \mathbb{P}\{h_{i+1} = n\}$$

and the proof is completed. ◇

Recall that, according to Definition 9, to achieve the same on the average energy dissipation *per area unit* (and thus per sensor) in the network area, the following equality should hold:

$$E\left[\frac{\sum_{k=1}^{h_i} \epsilon_{ik}}{S_i}\right] = E\left[\frac{\sum_{k=1}^{h_j} \epsilon_{jk}}{S_j}\right] \quad \forall i,j \in \{1,\ldots n\} \tag{5}$$

i.e. the average energy consumption per sensor should be equal in any two ring sectors. By induction, it suffices to guarantee this for any two adjacent sectors. In what follows, we guarantee the above balance property, requiring a certain recurrence relation to hold. This recurrence basically relates 3 successive terms of the $E[f_i]$ sequence (the $E[g_i]$ terms depend only on i and on input parameters).

Theorem 2 To achieve energy balance in the network, the following recurrency equation should hold:

$$a_{i+1}E[f_{i+1}] - (d_i + a_i)E[f_i] + d_{i-1}E[f_{i-1}] =$$
$$= a_i E[g_i] - a_{i+1}E[g_{i+1}]$$

where

$$a_i = \frac{i^2}{2i-1} \qquad d_i = \frac{(i+1)^2 - 1}{2i+1}$$

Proof: For the case $j = i + 1$ of the equation 5 and using the lemmata 7 and 9 we have:

$$\frac{E[h_i]E[\epsilon_{i,j}]}{S_i} = \frac{E[h_{i+1}]E[\epsilon_{i+1,j}]}{S_{i+1}} \quad \Leftrightarrow$$

$$\frac{E[h_i][i^2 - p_i(i^2-1)]}{(2i-1)} = \frac{E[h_{i+1}]\{(i+1)^2 - p_{i+1}[(i+1)^2 - 1]\}}{(2i+1)} \quad \Leftrightarrow$$

$$\frac{i^2}{2i-1}E[h_i] - p_iE[h_i]\frac{i^2-1}{2i-1} =$$

$$= \frac{(i+1)^2}{2i+1}E[h_{i+1}] - p_{i+1}E[h_{i+1}]\frac{(i+1)^2 - 1}{2i+1}$$

Let a_i, d_i as defined in the Theorem statement above. By lemma 11 we know that $p_iE[h_i] = E[f_{i-1}]$ and by lemma 10 it is $E[h_i] = E[g_i] + E[f_i]$ thus the last equation becomes:

$$a_{i+1}E[f_{i+1}] - (d_i + a_i)E[f_i] + d_{i-1}E[f_{i-1}] =$$

$$= a_i E[g_i] - a_{i+1}E[g_{i+1}]$$

◇

To solve the above recurrence we must compute $E[g_i]$.

Lemma 12 If N is the total number of events that are generated in the network, the mean value of g_i is given by the following relationship:

$$E[g_i] = N \cdot \lambda_i$$

Proof: Because the position of each event is independent of other events and because for each sector i, probability λ_i is the same, clearly g_i is binomial with parameters N, λ_i. ◇

In order to have a simpler recurrence involving only two (successive in fact) terms of the $E[f_i]$ sequence, we will transform the recurrence relation of Theorem 2 into the following (easier to solve) relation:

Lemma 13 The recurrency relation:

$$t_i - t_{i-1} = a_i \cdot E[f_i] - a_{i+1} \cdot E[f_{i+1}]$$
$$\text{for} \quad i = 1, \ldots n - 1$$
$$\text{and} \quad t_0 = a_1 \cdot E[f_1]$$

has as a solution the function

$$t_i = \sum_{j=1}^{i} (a_j E[g_j] - a_{j+1} E[g_{j+1}]) + a_1 \cdot E[f_1]$$

Proof: The proof is done by induction on i. For $i = 0$, it is obviously true. Let it be true for $i - 1$. For i we have:

$$t_i = t_{i-1} + a_i \cdot E[g_i] - a_{i+1} \cdot E[g_{i+1}]$$

By the induction hypothesis we get the solution

$$t_i = \sum_{j=1}^{i} (a_j E[g_j] - a_{j+1} E[g_{j+1}]) + a_1 \cdot E[f_1]$$

$\diamond$

Now the recurrency relation of Thrm 2 is simplified:

$$a_{i+1} \cdot E[f_{i+1}] - d_i \cdot E[f_i] = t_i \quad i = 1, \ldots, n - 1$$

Thus, we get a recurrence for sequence $E[f_i]$ involving only two successive terms of the sequence:

Theorem 3 The recurrency relation

$$a_{i+1} E[f_{i+1}] - d_i E[f_i] = t_i \quad i = 1, \ldots n - 1$$

where t_i is defined in lemma 13, has the following solution

$$E[f_{n-i}] = -\sum_{k=1}^{i} \frac{\prod_{j=k}^{i-1} a_{n-j}}{\prod_{j=k}^{i} d_{n-j}} \cdot t_{n-k}$$

The proof is rather complex, so we omit it here. The interested reader may find it in [16].

The full expression for $E[f_i]$ can be expressed by substituting i with $n - i$, thus

$$E[f_i] = -\sum_{k=1}^{n-i} \frac{\prod_{j=k}^{n-i+1} a_{n-j}}{\prod_{j=k}^{n-i} d_{n-j}} \cdot \left(\sum_{j=1}^{n-k} (a_j E[g_j] - a_{j+1} E[g_{j+1}]) + a_1 \cdot E[f_1] \right)$$

where $\prod_{i}^{i-1} a_i = 1$.

We note that all the parameters of the recurrency solution above are expressed as a function of $E[f_1]$ and i. So as to compute them, we firstly compute the value of $E[f_1]$. Then we can compute all the other parameters by replacing the already computed $E[f_1]$.

Now, the calculation of the probabilities p_i is quite easy.

Theorem 4 The energy balance property is achieved if any ring sector (say T_i) propagates each message it handles with probability p_i to the next ring sector, T_{i-1}, and with probability $1 - p_i$ it propagates the message directly to the sink. The value of each p_i is given by the following relation

$$p_i = \frac{E[f_{i-1}]}{E[g_i] + E[f_i]}$$

where the values of $E[f_i]$ and $E[g_i]$ are obtained from lemma 3 and lemma 12, respectively.

Proof: From equation 11 we know that $E[f_{i-1}] = p_i E[h_i]$ and also by lemma 10 we know that $E[h_i] = E[g_i] + E[f_i]$.   ◇

Remark. Note that, interestingly, p_i's are *independent* of the number N of the events that occur in the network, since p_i's depend only on i and the number of ring sectors n (which is broadcast to sectors by the sink). Thus the protocol assumes only local information.

We note that the analysis above allows the exact derivation of probabilities p_i's as a function of i and n which (although complicated and not obviously leading to a closed form) can be easily calculated by the sensors in the network by carrying out very simple calculations.

The authors of [16] also prove the correctness of the protocol:

Theorem 5 Given that the energy is each time on the average the same in all network sensors, each message will finally get to the sink.

4.5.   A Closed Form

Under specific assumptions (that we discuss and motivate) we can make the calculation of probabilities p_i simpler. Combining lemma 10 and lemma 12 we have that

$$E[h_i] = \lambda_i N + E[f_i] = \frac{2i - 1}{n^2} N + E[f_i] \tag{6}$$

By the corresponding relation for $E[h_{i-1}]$ it must be:

$$\frac{\frac{2i-1}{n^2}N + E[f_i]}{(2i-1)\frac{\phi}{2}R^2}[p_i + (1 - p_i)i^2]R^2 = \frac{\frac{2i-3}{n^2}N + E[f_{i-1}]}{(2i-3)\frac{\phi}{2}R^2}[p_{i-1} + (1 - p_{i-1})(i - 1)^2]R^2$$

But $2i - 1 \simeq 2i - 3$ and $\frac{\phi}{2}$, R^2 cancel. Dividing by N we get:

$$\left[1 + \frac{E[f_i]}{N}\right][p_i + (1 - p_i)i^2] \simeq \left[1 + \frac{E[f_{i-1}]}{N}\right][p_{i-1} + (1 - p_{i-1})(i - 1)^2]$$

If $E[f_i] \simeq E[f_{i-1}]$ then the previous relation becomes:

$$p_i + (1 - p_i)i^2 = p_{i-1} + (1 - p_{i-1})(i - 1)^2 \tag{7}$$

In [16] we show how to solve the above recurrence.

Theorem 6 If $E[f_i] \simeq E[f_{i-1}]$, $3 \le i \le n$, then the one-hop forwarding probability, guaranteeing energy balance, is

$$p_i = 1 - \frac{3x}{(i + 1)(i - 1)}$$

where $p_2 = x \in (0, 1)$ a free parameter and $p_1 = 0$.

We remark that the assumption $E[f_i] \simeq E[f_{i-1}]$ is quite reasonable and well motivated. We provide the following intuitive explanation of why this happens. Remark indeed that the area sizes of adjacent sectors (and thus the number of events generated in such sections) are more or less the same, especially when i increases. Furthermore, the probability p_i of forwarding to the adjacent (towards the sink) sector increases very fast with i and becomes 1 in most sectors of the network (in the middle territory).

4.6. Further research on energy balance

In [59] the authors propose a generalized data propagation algorithm, which allows to jump over bottleneck nodes. They investigate whether hops to intermediate slices help. They consider mixed propagation (i.e. only single hops and direct to sink transmissions) and interestingly show that mixed strategies beat every other possible strategy (wrt optimizing the network lifespan). Indeed, they prove that lifespan is maximized by a mixed data propagation algorithm. They then use this fact to propose am optimal distributed data propagation algorithm. Their method uses a LP maximizing the generalized flow, subject to certain constraints (detection rates, energy and flow constraints). They rigorously prove that if there exists an energy-balanced mixed flow it maximizes the mixed-flow and it also maximizes the generalized-flow. Thus, they can maximize the generalized-flow problem using a mixed-flow and a local property (energy-balance). Their algorithm produces a mixed flow and at the same time balances energy, while its stability is shown using Markov Chain methods.

In [58] the authors consider that nodes can send messages directly to the sink. They propose an algorithm to compute the optimal solution (in terms of lifetime), i.e. to compute the ejection probabilities. They interestingly show that when there exists an energy balanced solution, it is optimal. When it is not possible to balance energy, the algorithm still finds an optimal solution. They give a necessary and sufficient condition for a solution to be optimal.

Part B: Network Design for Energy Optimization

5. Tracking Moving Entities

5.1. Problem Description

We wish to solve the problem of *localizing and continuously tracking mobile objects* moving in a domain described by a set of set of three-dimensional curves, S, over a period of time T i.e. we want the wireless sensor network to be able to detect the position of any moving object, at any time t in T. We allow multiple targets that arise in the network area at random locations and at random times. The movement of each target can follow an arbitrary but continuous path i.e. we disallow the target to instantaneously "jump" to another location; still, we can handle such discontinuities as multiple targets.

In our setting, the set S of 3D curves is the set of possible trajectories of objects moving for some time within the period T. Such a moving object might follow a part of a curve in S, and possibly arrive to an intersection of curves and then follow another curve.

We can assume that each such curve in S is specified (for $t = 0, \ldots, T$) by an analytic equation (e.g. in x, y, z coordinates). Thus, we can compute a curve (route) *piece* τ_{ij} for the curve (route) R_i, via some criterion (e.g. to split R_i into pieces of equal length). We, then, wish to deploy some sensors (either

standing or even moving for some time) in some (initial) places within S. A basic demand for such a deployment consists of the following rule:

(Rule R) For every point $\vec{p} = (x, y, z)$ in S and for every $t \in [0, T]$ there are at least three sensors, active at time t, whose sensing range includes the point $\vec{p}$. Here, for a sensor σ operating at time t, its *sensing range* is a sphere of some radius $R_\sigma(t)$ and with center the position of σ at time t.

Note that rule R guarantees localization of any moving target in S at any time t in T, via triangulation.

We follow a *network design approach* to this problem. Let us assume, hypothetically, that we could have an abundance of sensors, each characterized by an initial position, an operation period, and an initial available energy (battery) that allows a particular implementation of sensing ranges during the operation of the sensor. Then, the decision to actually select one sensor (with the initial position, operating period and available energy) has a certain *cost*. Our goal is to be able to select a subset of sensors that implement rule (R) and is of nearly minimal cost.

The problem we study is related (but different) to the problems of network coverage and tracking. In fact, we extend the well-studied coverage problems by being able to track the moving path, and by also taking time into account. On the other hand, to reduce the energy dissipation and overhead of our tracking solution, we avoid some of the collaborative information processing components (like which nodes should sense, which have useful information and should communicate, which should receive information and how often). Thus, we are not dealing directly with queries of the type "how many targets are in a certain region during a certain time interval". Still, our solution performs the collaborative processing tasks (triangulation by at least three sensors) that allow localization of the targets as they move in the network. Also, we discuss how alternative collaborative processing methods can be combined with our approach to provide full tracking.

5.2. Our Contribution

Our approach indeed tries to avoid the expensive, massive placement of all the time functioning sensors all over the monitoring area, *by exploiting possible overlaps of routes* at certain places in the network area, using sensors that can simultaneously monitor pieces of several ("nearby") routes during their operation, or even exploit sensors on top of certain moving objects of our own that wander in the maze of routes. Thus, we somehow "multiplex" (both over space and time) the use of deployed sensors, since our approach identifies overlaps over space and time and thus deploys fewer sensors compared to the trivial approach.

In fact, the analysis of our method shows that *it is very efficient*, especially in very large domains, both with respect to computational time and the deployment cost, since it finds in polynomial expected time a deployment solution which approximates the optimal solution within a logarithmic factor. Thus, we avoid expensive dense deployment of sensors, where the information about the target is simultaneously generated by multiple sensors; we instead achieve a low cost solution (by removing unnecessary redundancy) that still keeps tracking accuracy at high levels.

To be able to handle overlaps, *we propose a novel combinatorial model for possible routes* in the network domain. This model, although abstract, captures several of the technical specifications of real sensor devices, such as the energy spent as a function of the transmission range, the ability to vary this range to save energy or increase connectivity, the ability of sensors to employ power saving (sleep-awake) schemes to save energy *etc.* This combinatorial model allows us to reduce the tracking problem

to a variation of a combinatorial problem of set covering (in particular to "at least 3 cover", i.e. having each point of the domain covered at any time by at least 3 sensors, and thus being localized). We feel that this combinatorial model is of independent interest and can (itself or its variations) be used in modeling other problems as well.

5.3.   Related Work and Comparison

As discussed in the problem definition part, the problem we study is relevant to network coverage and tracking, that we discuss below.

Coverage

Sensor deployment strategies play a very important role in providing better QoS, which relates to the issue of how well each point in the sensing field is covered. However, due to severe resource constraints and hostile environmental conditions, it is nontrivial to design an efficient deployment strategy that would minimize cost, reduce computation, minimize node-to-node communication, and provide a high degree of area coverage.

Several deployment strategies have been studied for achieving an optimal sensor network architecture. Dhilon *et al.* propose a grid coverage algorithm that ensures that every grid point is covered with a minimum confidence level [13]. They consider a minimalistic view of a sensor network by deploying a minimum number of sensors on a grid that would transmit a minimum amount of data. Their algorithm is iterative and uses a greedy heuristic to determine the best placement of one sensor at a time. It terminates when either a preset upper limit on the number of sensors is reached or sufficient coverage of the grid points is achieved. However, the algorithm assumes line of sight of the target and the sensor. Also, since a complete knowledge of the terrain is assumed, the algorithm is not very applicable in cluttered environments, such as interior of buildings, because modeling obstacles becomes extremely difficult in those scenarios. Finally, a main difference of this approach (and in fact the other coverage methods as well) with ours, is that we explicitly take time into account.

In contrast to static sensor networks, nodes in mobile sensor networks are capable of moving in the sensing field. Such networks are capable of self-deployment starting from an initial configuration. The nodes would spread out such that coverage in the sensing field is maximized while maintaining network connectivity. A potential field-based deployment approach using mobile autonomous robots has been proposed to maximize the area coverage [40]. Clearly, the assumptions of this method (and the one described below) are different to ours, since we do not use sensor mobility as an algorithmic design element (although we handle mobility when it appears).

Similar to the potential field approach, a sensor deployment algorithm in the presence of mobility based on virtual forces has been proposed in [53] to increase the coverage after an initial random deployment. A sensor is subjected to forces, which are either attractive or repulsive in nature. In this approach, obstacles exert repulsive forces, while areas of preferential coverage (sensitive areas where a high degree of coverage is required) exert attractive forces, and other sensors exert attractive or repulsive forces. A hard threshold distance is defined between two sensors to control how close they can approach each other.

Other interesting network coverage approaches are discussed in the book chapter by A. Ghosh and S.

Das in [43].

Tracking

Our method follows and extends the well-established line of research for a network architecture design for centralized placement/distributed tracking (see e.g. the book [43] for a nice overview). According to that approach, optimal (or as efficient as possible) sensor deployment strategies are proposed to ensure maximum sensing coverage with minimal number of sensors, as well as power conservation in sensor networks.

Centralized Approaches

In one of the methods, that focuses on deployment optimization, a grid manner discretization of the space is performed [7]. Their method tries to find the grid point closest to the target, instead of finding the exact coordinates of the target. In such a setting, an optimized placement of sensors will guarantee that every grid point in the area is covered by a unique subset of sensors. Thus, the sensor placement problem can be modeled as a special case of the alarm placement problem described by Rao [41]. That problem is the following: given a graph G, which models a system or a network, one must determine how to place "alarms" on the nodes of G so that any single node fault can be diagnosed. It has been shown in [41] that the minimal placement of alarms for arbitrary graphs is an NP-complete problem. Clearly, their problem is easier than ours, since they relax the requirement to find exact coordinates of moving objects by just finding the nearest grid point. Since their problem is computationally difficult, this implies the inherently high complexity of our problem. Another indication of the hardness of the problem is the fact that, as shown in [5], the localization problem is NP-hard in sparse wireless sensor networks.

Also, our problem is related but different to the following other well known approach that focuses on power conservation: in [18] sleep−awake patterns for each sensor node are obtained during the tracking stage, to obtain power efficiency. The network operates in two stages: the surveillance stage during the absence of any event of interest, and the tracking stage, in response to the presence of moving targets. Each sensor initially works in the low-power mode when there are no targets in its proximity. However, it should exit the low-power mode and be active continuously for a certain amount of time when a target is sensed, or even better, when a target is shortly about to enter. Finally, when the target passes by and moves farther away, the node should decide to switch back to the low-power mode. Our approach is also power aware in the same sense (since we also affect the duration of sensors' operation), but additionally we also control the transmission range (and thus the power dissipation).

Another centralized approach, is "sensor specific", in the sense it uses some smart powerful sensors that have high processing abilities [19]. In particular, this algorithm assumes that each node is aware of its absolute location via a GPS or a relative location. The sensors must be capable of estimating the distance of the target from the sensor readings.

Distributed Approaches

As opposed to centralized processing, in a distributed model sensor networks distribute the computation among sensor nodes. Each sensor unit acquires local, partial, and relatively coarse information from

its environment. The network then collaboratively determines a fairly precise estimate based on its coverage and multiplicity of sensing modalities. Several such distributed approaches have been proposed. Although we are not comparing with them, we shortly discuss some of them, for completeness.

In [32], a cluster-based distributed tracking scheme is provided. The sensor network is logically partitioned into local collaborative groups. Each group is responsible for providing information on a target and tracking it. Sensors that can jointly provide the most accurate information on a target (in this case, those that are nearest to the target) form a group. As the target moves, the local region must move with it; hence groups are dynamic with nodes dropping out and others joining in. It is clear that time synchronization is a major prerequisite for this approach to work. Furthermore, this algorithm works well for merging multiple tracks corresponding to the same target. However, if two targets come very close to each other, then the mechanism described will be unable to distinguish between them.

Another nice distributed approach is the dynamic convoy tree-based collaboration (DCTC) framework that has been proposed in [51]. The convoy tree includes sensor nodes around the detected target, and the tree progressively adapts itself to add more nodes and prune some nodes as the target moves. In particular, as the target moves, some nodes lying upstream of the moving path will drift farther away from the target and will be pruned from the convoy tree. On the other hand, some free nodes lying on the projected moving path will soon need to join the collaborative tracking. As the tree further adapts itself according to the movement of the target, the root will be too far away from the target, which introduces the need to relocate a new root and reconfigure the convoy tree accordingly. If the moving target's trail is known a priori and each node has knowledge about the global network topology, it is possible for the tracking nodes to agree on an optimal convoy tree structure; these are at the same time the main weaknesses of the protocol, since in many real scenaria such assumptions are unrealistic.

The interested reader is encouraged to refer to [52], the nice book by F. Zhao and L. Guibas, that even presents the tracking problem as a "canonical" problem for wireless sensor networks. Also, several tracking approaches are presented in [43]. Other tracking methods, including interesting algorithmic components, have appeared in [53, 55–57]. This work has appeared in [37].

5.4. *The Model and its Combinatorial Abstraction*

Sensors and Sensor Network

The basic model M_0 is used, with some additional assumptions (see section 1.3). Let n be the total number of *available* sensor devices for deployment. Let S be the set of *possible* deployment positions (i.e. the union of 3D curves as we described earlier). Let T be a period of time. For each sensor σ of the n available sensors, a placement act, A_σ, is a decision (i) either not to deploy σ or (ii) to deploy σ in an initial position $\vec{p}$ in S, for a period $T_\sigma \subseteq T$, with a pre-specified pattern of $R_1(t), R_2(t)$ ($t \in T_\sigma$) and a possible trajectory of σ moving in S for all t in T_σ. All the placement acts, together, form a sensor network N. We assume here that N is capable of (somehow) reporting the sensing of local events (i.e. tracking events) to some set of sinks T_N.

A model for targets

In contrast to models that allow only a single moving target, we allow *multiple* targets. We assume that *the initial positions of all targets are arbitrary*. Also, we assume that *all targets arise in arbitrary*

times during the network operation.

With respect to target mobility, we assume that each target follows an arbitrary path in S which is however continuous i.e. we disallow the target to instantaneously "jump" to another location. Furthermore, *we do not limit the movement speed of targets*; we only assume that when a target enters the sensing area of an (awake) sensor, it does not manage to leave this area before being sensed. This limit on motion speed is rather trivial, since sensing speed is very high i.e. practically the time needed for a target to be sensed is very close to zero.

The combinatorial model

Assume a given complicated 3D domain, S, with obstacles that disallow signal transmissions (e.g. a set of corridors in buildings or different shape obstacles in a mountain). By "complicated" we mean that the domain can be represented by *arbitrary* three-dimensional curves (see also the problem description subsection), i.e. the only modeling restriction on the curves is their continuity. We further assume that the domain can be represented by the union of λ routes $R_1, \ldots, R_\lambda$ (each can be realized by e.g. a moving robot or air-vessel). We are also given a period T of time.

We wish to equip each route with sensors (of varying capabilities e.g. varying transmission range and operation times) so that any moving object in the domain at any time t in T can be "seen" by at least three sensors (and, thus, its instantaneous position can be found by triangulation). If we can manage this, we can monitor the motion of any moving object within S during T.

A trivial, but very costly, solution is to equip *each route* with sensors (in various points of the route) each operating during the whole T and being able to track any motion in a part of the route. However, one could *exploit overlaps of routes at certain places*, sensors that can monitor pieces of several routes ("nearby") during their operation, or even sensors on top of certain moving objects of our own that wander in the maze of routes.

In order to argue about such economic tracking methods, we view each R_i partitioned into several "route pieces" r_{ij}. Let n_i be the number of the pieces of R_i. We also partition the period T into suitable *intervals* $\tau_1, \ldots, \tau_k$ so that $T = \tau_1 + \cdots + \tau_k$.

We call an *"element"* each pair (r_{ij}, τ_m), for $1 \leq i \leq \lambda$, $1 \leq j \leq n_i$ and $1 \leq m \leq k$. There are $n = \left(\sum_{i=1}^{\lambda} n_i \right) \cdot k$ such elements.

We can then describe sensor placements (that work for a certain duration each) as relations (sets) between those elements. For example: (a) a sensor placed at r_{i5} can also "see" r_{j7}, r_{k30}. This sensor can operate for 3 intervals. If we start it at τ_1 then the element (r_{i5}, τ_1) "covers" the elements (r_{i5}, τ_2), (r_{i5}, τ_3) but also (r_{j7}, τ_1), (r_{j7}, τ_2), (r_{j7}, τ_3) and (r_{k30}, τ_1) (r_{k30}, τ_2) (r_{k30}, τ_3). (b) A sensor is attached to a moving object that moves from r_{11} to r_{12} to r_{13} and then r_{34}, r_{35}, r_{36}. The sensor lasts 6 intervals and our moving object starts at τ_5. Then, element (r_{11}, τ_5) "covers" elements (r_{12}, τ_6), (r_{13}, τ_7) and also (r_{34}, τ_8), (r_{35}, τ_9), (r_{36}, τ_{10}) (and itself, of course).

In general, each placement of a certain sensor activated at a certain time and operating for a certain time, corresponds to a *set of "covered" elements*. Each such "set" has *a certain cost* e.g. it is more expensive if the sensor's battery is such that the sensor lasts for a long time. Also it is more expensive if its sensing range is large and can "see" more route pieces.

Definition 12 A redundant monitoring design (RMD) D is a set of possible choices of $q \leq n$ sensors

$\sigma_1, \ldots, \sigma_q$, each with a placement act $A(\sigma_i)$ of the form "deploy" and the associated cost of the placement act.

Each RMD results in a family of sets $\Sigma_1, \ldots, \Sigma_q$, each having a cost $c(\Sigma_i) > 0$ and each being a subset of our universal set of elements $U = \{e : e \text{ is a pair } (r_{ij}, \tau_m)\}$.

For *feasibility* of the RMD we can require:

(a) that the union of all Σ_i is U.

(b) that each e in U belongs to at least 3 Σ_i sets initially.

However, this is not necessary for our method, since the method itself will discover an infeasible RMD.

Definition 13 Given are an instance of an RMD of sets $\Sigma_1, \ldots, \Sigma_q$ on the universe U of elements related to the domain S and the period T, and also a cost $c(\Sigma_i) \geq 0$ for each Σ_i. Then, an *optimal* final monitoring decision (Optimal FMD) is a sub-collection, F, of sets $\{\Sigma_{t_1}, \ldots, \Sigma_{t_{q'}}\}$ (where $q' \leq q$ and each $t_i \in \{1, 2, \ldots, q\}$), whose total cost $c(F) = \sum_{j=1}^{q'} c(\Sigma_{t_j})$ is minimum, and such that (i) $\cup_i \sum_{t_i} = U$ (ii) each e in U belongs to at least 3 sets in the sub-collection F.

We note that we can construct several RMDs for each run of our Algorithm, get a close to optimal solution (FMD) for each and select the best among them.

5.5. A way to compute near optimal FMDs

From the above formulation, the Optimal FMD problem is actually the following "AT-LEAST-3-SET-COVER" problem:

AT-LEAST-3-SET-COVER ($\geq 3SC$): Given $D = (\Sigma_1, \ldots, \Sigma_q)$ where each $\Sigma_i \subseteq U$ (and the union of all Σ_i is U) and given the costs $c(\Sigma_i) \geq 0$, select a minimum total cost sub-collection F of D so that each element e in U belongs to (is "covered" by) at least 3 sets in F.

Important note: Note that geometry and geometric covers can not help here because the domain since S is a complicated 3D domain that can be highly irregular; also, the timing parts of the elements escape the Cartesian geometry; finally, the cost of each Σ_i is a complicated function of placement decisions of sensors of various capabilities.

The problem of $\geq 3SC$ is NP-hard. This is so, since the usual min-cost SET-COVER problem can be reduced to it by adding to any instance of SET-COVER two sets of zero cost, each covering all elements.

We now describe a formulation of the problem as an integer linear program and give an approximate solution based on randomized rounding. Note that our method can be extended to low cost "SET-COVER by a at least l sets" (let us denote this problem as $\geq l$-SET-COVER) if the redundancy of $l > 3$ is needed to make the *redundant monitoring decision D* easier to construct and fault-tolerant.

We remind the reader of the computational complexity of set covering problems: Lund and Yannakakis showed in 1994 that set cover cannot be approximated in polynomial time to within a logarithmic factor, unless NP has quasi-polynomial time algorithms [33]. Feige improved their inapproximability lower bound, under the same assumptions, giving a slightly better bound that essentially matches the approximation ratio achieved by the greedy algorithm [17]. Alon, Moshkovitz, and Safra established a larger lower bound, under the weaker assumption that P $\neq$ NP [2]. The k-set cover problem is a variant in which every set is of size bounded by k. While k-set cover problem can be solved in polynomial

time (via matchings) for $k = 2$, it is NP-complete and even MAX SNP-hard for $k \geq 3$. Greedy algorithms in that case achieve an approximation ratio of $\ln k + \Theta(1)$. Hardness results show that it is not approximable within $(1 - \epsilon) \ln n$, under strong complexity-theoretic evidence [17]. For 3-set cover, besides linear programming techniques (fractional covers), also local and "semi-local" approximation techniques have been used e.g. in [14, 20, 21].

The randomized rounding method

For set Σ let $x(\Sigma)$ be 1 if Σ is selected and 0 else. We want to

$$\text{minimize} \sum_{\Sigma \text{ in } D} x(\Sigma) \cdot c(\Sigma)$$

subject to
(1) $x(\Sigma) \in \{0, 1\}$
(2) Let $E(e)$ be the collection of all Σ containing element e. Then, for each element e,

$$\sum_{\Sigma \text{ in } E_e} x(\Sigma) \geq 3$$

Let IP1 be the above integer linear program.
We relax the above integer program to the following linear program (LP1):

$$\text{minimize} \sum_{\Sigma \text{ in } D} x(\Sigma) c(\Sigma)$$

given that
(1)$\forall \Sigma, 0 \leq x(\Sigma) \leq 1$
(2) $\forall e, \sum_{\Sigma \text{ in } E(e)} x(\Sigma) \geq 3$

Let $\{p(\Sigma)\}$ be the optimal solution to the above (i.e. $x(\Sigma) = p(\Sigma)$ get the minimum). We can find $\{p(\Sigma)\}$ in polynomial time in the size n of the redundant monitoring decision D.

We then form a subcollection of sets as follows:

Initially $C = \emptyset$
Experiment E.
For each Σ in D, put Σ into the sub-collection C with probability $p(\Sigma)$, independently of the others.

The experiment E above outputs a sub-collection C. Clearly,

$$E(cost(C)) = \sum_{\Sigma \text{ in } D} c(\Sigma) \mathbb{P}\{\Sigma \text{ is picked}\} = \sum_{\Sigma \text{ in } D} c(\Sigma) p(\Sigma) = OPT_f$$

where OPT_f is the optimal value of the linear program LP1. The found collection C has a very nice expected cost (even better than what one can achieve in our original *integer* problem) but we have to examine feasibility.

Since we want to get a C where each e in U belongs to at least 3 sets of C, we must examine whether the (random) C obtained has this property.

Definition 14 *Let α be an element in U and C obtained by the experiment E. We denote by $p(\alpha, C)$ the probability that α belongs to at least 3 sets of C.*

Let w.l.o.g. $\Sigma_1, \ldots, \Sigma_\lambda$ be the sets of our RMD containing element α. Here, we must have $\lambda \geq 3$, else LP1 will report infeasibility. W.l.o.g. denote by p_i the probability $p\left(\sum_i\right)$ obtained via LP1, i.e. that Σ_i is chosen to be in C. Assuming that LP1 is feasible we get:

$$\gamma = p_1 + \cdots + p_\lambda \geq 3 \tag{8}$$

Let A_3, N_0, N_1, N_2 be the events:
$A_3 = $ "α is covered by at least 3 sets in C"
$N_0 = $ "α does not belong to any set in C"
$N_1 = $ "α belongs to exactly one set in C"
$N_2 = $ "α belongs to exactly two sets in C"
Now,

$$p(\alpha, C) = \mathbb{P}\{A_3\} = 1 - \mathbb{P}\{N_0\} - \mathbb{P}\{N_1\} - \mathbb{P}\{N_2\} \tag{9}$$

We now estimate $\mathbb{P}\{N_i\}$, for $i = 0, 1, 2$:
We will repeatedly use the following fact:

Fact (*) If the numbers $x_1, \ldots, x_\lambda$ are each in $[0, 1]$ and $x_1 + \cdots + x_\lambda \geq \gamma$, then $(1 - x_1) \cdots (1 - x_\lambda) \leq \left(1 - \frac{\gamma}{\lambda}\right)^\lambda \leq e^{-\gamma}$ (it is assumed that $\gamma \leq \lambda$).

Fact (*) can be proved via an easy induction, or via the fact that the arithmetic mean is bigger or equal than the geometric one.

Proof of Fact (*) Let $y_i = 1 - x_i, i = 1, \ldots, \lambda$. Then, $\sum y_i = \lambda - \sum x_i \leq \lambda - \gamma$. So,

$$\frac{\sum y_i}{\lambda} \leq 1 - \frac{\gamma}{\lambda} \tag{10}$$

But, $\frac{\sum y_i}{\lambda} \geq \sqrt[\lambda]{\pi y_i}$ (arithmetic mean vs geometric mean), so:

$$\frac{\sum y_i}{\lambda} \geq \sqrt[\lambda]{(1 - x_1) \cdots (1 - x_\lambda)}$$

So, by Equation (3), $\sqrt[\lambda]{(1 - x_1) \cdots (1 - x_\lambda)} \leq 1 - \frac{\gamma}{\lambda}$. So, $(1 - x_1) \cdots (1 - x_\lambda) \leq \left(1 - \frac{\gamma}{\lambda}\right)^\lambda$ $\diamond$

We now have:
(a) $\mathbb{P}\{N_0\} = (1 - p_1) \cdots (1 - p_\lambda)$. By the Fact (*), then

$$\mathbb{P}\{N_0\} \leq \left(1 - \frac{\gamma}{\lambda}\right)^\lambda \leq \exp(-\gamma)$$

.

Also, by inclusion-exclusion,

$$\mathbb{P}\{N_1\} = p_1 \cdot \mathbb{P}\{\text{no cover of } \alpha \text{ in the trials of } \Sigma_2, \ldots, \Sigma_\lambda\} +$$

$$+ (1 - p_1)\mathbb{P}\{1 \text{ cover of } \alpha \text{ in the trials of } \Sigma_2, \ldots, \Sigma_\lambda\} \leq p_1 \cdot \frac{1}{e^{\gamma - p_1}} + (1 - p_1)\mathbb{P}\{N_1\}$$

(because $p_2 + \cdots + p_\lambda = \gamma - p_1$ so $(1 - p_2) \cdots (1 - p_\lambda) \le \left(1 - \frac{\gamma - p_1}{\lambda}\right)^\lambda = \exp(-(\gamma - p_1))$ and also because $\mathbb{P}\{1 \text{ success in the trials of } \Sigma_2, \ldots, \Sigma_\lambda\} \le \mathbb{P}\{N_1\}$).

So,

$$\mathbb{P}\{N_1\} \le p_1 \cdot \frac{1}{e^{\gamma - p_1}} + (1 - p_1)\mathbb{P}\{N_1\}$$

so

$$\mathbb{P}\{N_1\} \le \frac{1}{e^{\gamma - p_1}} \le \frac{1}{e^{\gamma - 1}}$$

Similarly,

$$\mathbb{P}\{N_2\} = p_1 \cdot \mathbb{P}\{1 \text{ cover of } \alpha \text{ in the trials of } \Sigma_2, \ldots, \Sigma_\lambda\} +$$

$$+ (1 - p_1)\mathbb{P}\{2 \text{ covers of } \alpha \text{ in the trials of } \Sigma_2, \ldots, \Sigma_\lambda\} \le p_1 \cdot \frac{1}{e^{\gamma - p_1}} + (1 - p_1)\mathbb{P}\{N_2\}$$

also, so $\mathbb{P}\{N_2\} \le \frac{1}{e^{\gamma - 1}}$.

Thus,

$$\mathbb{P}\{A_3\} = p(a, c) \ge 1 - \frac{1}{e^\gamma} - \frac{2}{e^{\gamma - 1}}$$

and $\gamma \ge 3$. So, $\mathbb{P}\{A_3\} = p(a, c) \ge 1 - \frac{1}{e^3} - \frac{2}{e^2} > \frac{1}{2}$. Let $\xi = 1 - \frac{1}{e^3} - \frac{2}{e^2}$. Hence, we get the following:

Theorem 7

$$\mathbb{P}\{\alpha \text{ is covered by at least 3 sets in } C\} = p(\alpha, c) \ge 1 - \frac{1}{e^3} - \frac{2}{e^2} = \xi$$

We now repeat the experiment E (with the same $p(\Sigma_i)$) to get $r = c \log n$ such collections $C_1, C_2, \ldots, C_r$. Let $V = C_1 \cup C_2 \cup \ldots \cup C_r$. By independence of the repeated experiments, if the event AV_3 is : $AV_3 =$ "element α is not covered by at least 3 sets in V" then

$$\mathbb{P}\{AV_3\} \le (1 - \xi)^{c \log n}$$

We can always choose c so that $(1 - \xi)^c < 1/4$. Then,

$$\mathbb{P}\{AV_3\} \le \left(\frac{1}{4}\right)^{\log n} \le \frac{1}{n^2}$$

Thus the probability that there is an element in U not covered by at least 3 sets of V is bounded by above by $n \cdot n^{-2} = 1/n$.

So, we get the following:

Theorem 8 The collection V obtained satisfies: (i)

$$\mathbb{P}\{V \text{ covers each element by at least 3 sets}\} \le 1 - \frac{1}{n}$$

and (ii)

$$E(cost(V)) \le c \log n \, OPT_f \le c \log n \, OPT$$

where OPT is the cost of the *optimal* FMD.

Let $\rho = c \log n$. Note that by the Markov inequality it is

$$\mathbb{P}\{cost(V) < 2\rho \, OPT_f\} \geq \frac{1}{2}$$

Thus, the probability that V is valid and has cost less than $2\rho OPT_f$ is at least

$$1 - \left(\frac{1}{n} + \frac{1}{2}\right) = \frac{1}{2} - \frac{1}{n}$$

We can then repeat the whole process an expected number of at most 2 times and get a V which is verified to be an almost optimal and valid FMD. Note that we have also shown:

Theorem 9 The problem $\geq 3SC$ can be approximated in polynomial expected time with an approximation ratio $\Theta(\log n)$.

5.6. The triangulation issue

The solution to $\geq 3SC$ of the last section selects a close to optimal FMD. Thus, it also specifies the initial positions of the associated (selected) sensors. However, no guarantee is provided that for any element e, the 3 elements "covering" e actually form a triangle (e.g. they may be on a line). We propose to handle this via a "post-processing" step as follows:

For each $e = (r_{ij}, \tau_m)$ let $\sigma_1(e), \sigma_2(e), \sigma_3(e)$ be the 3 sensors covering e (i.e. covering r_{ij} at τ_m). We now modify their placement act by perturbing the positions p_i^* of $\sigma_i(e)$ at τ_m by a random, small perturbation of center their p_i^* and radius $\epsilon > 0$, small enough so that the same sets are covered by them. We perform the perturbation act only for those e for which the $\sigma_i(e)$ are not forming a triangle.

At the end of the post-processing step, each e in U is covered by (at least) 3 sensors, whose positions (during τ_m) form a triangle with high probability. We can repeat the perturbation until the triangle is indeed formed.

This post-processing step can always be done, provided that S gives an infinitesimal free space around each of its points. This is safe to assume for any application. We note that our solution works for static sensors, and moving sensors whose motion is controlled by us, while it can not be applied to the case of sensors moving on their own.

5.7. Alternative collaborative processing methods for our approach

Our method is based on an easy to get RMD which is then "cleared" via the randomized rounding technique to get a low cost final monitoring decision. Its result is a selection of sensor placement acts, guaranteed to monitor each point $\vec{p}$ in S by at least 3 sensors for any time in the period T. When our localization method is combined with ability to distinguish all observed objects (e.g. by using unique IDs) then tracking of objects can be performed.

Clearly our method must be complemented by a way to report target positions and their associated times to some central facility. If the reporting delay is comparable to the speed of the target then the central facility can reconstruct the target's motion in real time. To this end, Delay and Disruption Tolerant Networking (DTN, see e.g. [28]) approaches can be useful, to improve network communication when connectivity is periodic, intermittent or prone to disruptions and when multiple heterogeneous underlying networks may need to be utilized to effect data transfers.

The sensor network may handle the localization reports (i.e. reports on target position at a certain time) distributedly. This reduces communication cost but the central facility must have a way to compare those reports into a consistent information about the trajectory of the target. The issues of local clocks and their synchronization is crucial for this and we do not solve it here. In fact, we view our approach as a building block for full tracking approaches.

6. Conclusions

We investigated several aspects of energy optimization in sensor networks, like minimizing the total energy spent in the network, minimizing the number (or the range) of data transmissions, combining energy efficiency and fault-tolerance (by allowing redundant data transmissions which however should be optimized to not spend too much energy), and balancing the energy dissipation among the sensors in the network, in order to avoid the early depletion of certain sensors and thus the breakdown of the network. Since it is very difficult (if possible at all) to achieve all the above goals at the same time we presented a variety of protocols, each of which may possibly focus at some of the energy efficiency goals above (while still performing well with respect to the rest goals as well). In particular, we presented three energy efficient protocols: (a) The Local Target Protocol (LTP), that performs a local optimization trying to minimize the number of data transmissions (b) The Probabilistic Forwarding Protocol (PFR), that creates redundant data transmissions that are probabilistically optimized, to trade-off energy efficiency with fault-tolerance and (c) The Energy Balanced Protocol (EBP), that focuses on guaranteeing the same per sensor energy dissipation, in order to prolong the lifetime of the network.

For the problem of how to design low cost sensor networks able to efficiently localize and track multiple moving targets, we tried to take advantage of possible coverage overlaps over space and time, by introducing an abstract, combinatorial model that captures such overlaps. Under this model, we abstracted the localization and tracking problem by a combinatorial problem of set covering (by at least three sets, to ensure localization of any point in the network area, via triangulation). We then provided an efficient approximate method for sensor placement and operation, that with high probability and in polynomial expected time achieves monitoring decisions with a $\Theta(\log n)$ approximation ratio to the optimal solution.

Open issues for further research include considering the impact on protocols' performance of other important types of faults (malicious, byzantine). Also, it is worth investigating other performance aspects like congestion (possibly using game-theoretic methods).

References and Notes

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: a survey. *J. Comput. Networks* **2002**, *38*, 393-422.
2. Alon, N.; Moshkovitz, D.; Safra, M. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms* **2006**, *2*, 153-177.
3. Antoniou, T.; Boukerche, A.; Chatzigiannakis, I.; Mylonas, G.; Nikoletseas, S. A New Energy Efficient and Fault-tolerant Protocol for Data Propagation in Smart Dust Networks using Varying Transmission Range. In Proc. 37th Annual ACM/IEEE Simulation Symposium (ANSS'04), 2004.
4. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A Survey on Sensor Networks. *IEEE*

Commun. Mag. **2002**, *40*, 102-114.

5. Aspnes, J.; Goldberg, D.; Yang, Y. On the computational complexity of sensor network localization. In Proc. 1st Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004); Springer-Verlag: Heidelberg, Germany, 2004; Lecture Notes in Computer Science, LNCS 3121, pp. 32-44.

6. Boukerche, A.; Cheng, X.; Linus, J. Energy-Aware Data-Centric Routing in Microsensor Networks. In Proc. of ACM Modeling, Analysis and Simulation of Wireless and Mobile Systems, September 2003, pp. 42-49.

7. Chakrabarty, K.; Iyengar, S.S.; Qi, H.; Cho, E. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Trans. Comput.* **2002**, *51*, 1448-1453.

8. Chatzigiannakis, I.; Nikoletseas, S.; Spirakis, P. Smart Dust Protocols for Local Detection and Propagation. In Proc. 2nd ACM Workshop on Principles of Mobile Computing (POMC'2002), 2002. This article also appeared as: Chatzigiannakis, I.; Nikoletseas, S.; Spirakis, P. Efficient and Robust Protocols for Local Detection and Propagation in Smart Dust Networks. *Mobile Networks and Applications* **2005**, *10*, 133-149.

9. Chatzigiannakis, I.; Dimitriou, T.; Nikoletseas, S.; Spirakis, P. A Probabilistic Algorithm for Efficient and Robust Data Propagation in Smart Dust Networks. In Proc. 5th European Wireless Conference on Mobile and Wireless Systems beyond 3G (EW 2004), 2004. This article also appeared as: Chatzigiannakis, I.; Dimitriou, T.; Nikoletseas, S.; Spirakis, P. A probabilistic algorithm for efficient and robust data propagation in wireless sensor networks. *Ad-Hoc Networks* **2006**, *4*, 621-635.

10. Chatzigiannakis, I.; Dimitriou, T.; Mavronicolas, M.; Nikoletseas, S.; Spirakis, P. A Comparative Study of Protocols for Efficient Data Propagation in Smart Dust Networks. In Proc. International Conference on Parallel and Distributed Computing (EUPOPAR), 2003; *Euro-Par 2003 Parallel Processing*; Springer-Verlag: Heidelberg, Germany, 2003; Lecture Notes in Computer Science, Capter 15, pp. 1003-1016. This article also appeared as: Chatzigiannakis, I.; Dimitriou, T.; Mavronicolas, M.; Nikoletseas, S.; Spirakis, P. A Comparative Study of Protocols for Efficient Data Propagation in Smart Dust Networks. *Parallel Processing Letters* **2003**, *13*, 615-627.

11. Chatzigiannakis, I.; Nikoletseas, S. A Sleep-Awake Protocol for Information Propagation in Smart Dust Networks. In Proc. 3rd Workshop on Mobile and Ad-Hoc Networks (WMAN), IPDPS Workshops; IEEE Press, 2003; p. 225. This article also appeared as: Chatzigiannakis, I.; Nikoletseas, S. Design and analysis of an efficient communication strategy for hierarchical and highly changing ad-hoc mobile networks. *Mobile Networks and Applications* **2004**, *9*, 319-332.

12. Chen, B.; Jamieson, K.; Balakrishnan, H.; Morris, R. SPAN: An energy efficient coordination algorithm for topology maintenance in ad-hoc wireless networks. In Proc. ACM/IEEE International Conference on Mobile Computing (MOBICOM), 2001.

13. Dhilon, S.S.; Chakrabarty, K.; Iyengar, S.S. Sensor placement for grid coverage under imprecise detections. In Proc. 5th International Conference on Information Fusion, (FUSION'02), Annapolis, MD, Jul 2002; pp. 1-10.

14. Duh, R.-C.; Fuerer, M. Approximation of k-Set Cover by Semi-Local Optimization. In Proc. 29th Annual ACM Symposium on the Theory of Computing, 1997; Association for Computer Machinery: Nwe York, NY, USA, 1997; pp. 256–264.

15. Estrin, D.; Govindan, R.; Heidemann, J.; Kumar, S. Next Century Challenges: Scalable Coordination in Sensor Networks. In Proc. 5th ACM/IEEE International Conference on Mobile Computing, (MOBICOM'99), 1999.

16. Euthimiou, H.; Nikoletseas, S.; Rolim, J. Energy Balanced Data Propagation in Wireless Sensor Networks. In Proc. 4th International Workshop on Algorithms for Wireless, Mobile, Ad-Hoc and Sensor Networks, (WMAN '04), 2004. This article also appeard as: Euthimiou, H.; Nikoletseas, S.; Rolim, J. Energy balanced data propagation in wireless sensor networks. *Wireless Networks* **2006**, *12*, 691-707.

17. Feige, U. A Threshold of $\ln n$ for Approximating Set Cover. *J. ACM* **1998**, *45*, 634-652.

18. Gui, C.; Mohapatra, P. Power conservation and quality of surveillance in target tracking sensor networks. In Proc. ACM MobiCom Conference, 2004.

19. Gupta, R.; Das, S.R. Tracking moving targets in a smart sensor network. In Proc. VTC Symp., 2003.

20. Halldorsson, M.M. Approximating Discrete Collections via Local Improvements. In Proc. ACM SODA, 1995.

21. Halldorsson, M.M. Approximating k-Set Cover and Complementary Graph Coloring. In Proc. IPCO, 1996.

22. Hollar, S.E.A. COTS Dust; M.Sc. Thesis, University of California, Berkeley, USA, 2000.

23. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In Proc. 33rd Hawaii International Conference on System Sciences (HICSS'2000), 2000.

24. Heinzelman, W.R.; Kulik, J.; Balakrishnan, H. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In Proc. 5th ACM/IEEE International Conference on Mobile Computing (MOBICOM'1999), 1999.

25. Intanagonwiwat, C.; Govindan, R.; Estrin, D. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In Proc. 6th ACM/IEEE International Conference on Mobile Computing (MOBICOM'2000), 2000.

26. Intanagonwiwat, C.; Govindan, R.; Estrin, D.; Heidemann, J.; Silva, F. Directed Diffusion for Wireless Sensor Networking. *IEEE/ACM Trans. Networking* **2003**, *11*, 2-16. (Extended version of [25]).

27. Intanagonwiwat, C.; Estrin, D.; Govindan, R.; Heidemann, J. Impact of Network Density on Data Aggregation in Wireless Sensor Networks. Technical Report 01-750, University of Southern California Computer Science Department, November, 2001.

28. Jain, S.; Demmer, M.; Patra, R.; Fall, K. Using Redundancy to Cope with Failures in a Delay Tolerant Network. In Proc. SIGCOMM, 2005.

29. Kahn, J.M.; Katz, R.H.; Pister, K.S.J. Next Century Challenges: Mobile Networking for Smart Dust. In Proc. 5th ACM/IEEE International Conference on Mobile Computing, September 1999; pp. 271-278.

30. Karp, B. Geographic Routing for Wireless Networks. Ph.D. Thesis, Harvard University, Cambridge, USA, 2000.

31. Kleinrock, L. *Queueing Systems Theory*; John Wiley & Sons: New York, NY, USA, 1975; Vol. I, p. 100.

32. Liu, J.; Liu, J.; Reich, J.; Cheung, P.; Zhao, F. Distributed group management for track initiation and maintenance in target localization applications. In Proc. Int. Workshop on Information Processing in Sensor Networks (IPSN), 2003.

33. Lund, C.; Yannakakis, M. On the hardness of approximating minimization problems. *J. ACM*, 1994, 960-981.

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

35. Manjeshwar, A.; Agrawal, D.P. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In Proc. 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, satellite workshop of the 16th Annual International Parallel & Distributed Processing Symposium (IPDPS'02), 2002.

36. Mehlhorn, K.; Näher, S. *LEDA: A Platform for Combinatorial and Geometric Computing*; Cambridge University Press: Cambridge, UK, 1999.

37. Nikoletseas, S.; Spirakis, P. Efficient Sensor Network Design for Continuous Monitoring of Moving Objects. In Proc. of the Third International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 07); Springer-Verlag: Heidelberg, Germany, 2008; Lecture Notes in Computer Science (LNCS), Volume 4837, pp. 18-31. This article also appeared in: *Theoretical Computer Science* **2008**, *402*, 56-66.

38. Nikoletseas, S.; Chatzigiannakis, I.; Euthimiou, H.; Kinalis, A.; Antoniou, A.; Mylonas, G. Energy Efficient Protocols for Sensing Multiple Events in Smart Dust Networks. In Proc. 37th Annual ACM/IEEE Simulation Symposium (ANSS'04), 2004.

39. Perkins, *C.E. Ad Hoc Networking*; Addison-Wesley: Boston, MA, USA, 2001.

40. Poduri, S.; Sukhatme, G.S. Constrained coverage in mobile sensor networks. In Proc. IEEE Int. Conf. Robotics and Automation (ICRA'04), New Orleans, LA, USA, April-May 2004; pp. 40-50.

41. Rao, N.S.V. Computational complexity issues in operative diagnostics of graph based systems. *IEEE Trans. Comput.* **1993**, *42*; pp. 447-457.

42. Ross, S.M. *Stochastic Processes*, 2nd Edition; John Wiley and Sons: Hoboken, NJ, USA, 1995.

43. Shorey, R.; Ananda, A.; Chan, M.C.; Ooi, W.T. Mobile, Wireless, and Sensor Networks: Technology, Applications, and Future Directions. *Wiley* 2006.

44. Schurgers, C.; Tsiatsis, V.; Ganeriwal, S.; Srivastava, M. Topology Management for Sensor Networks: Exploiting Latency and Density. In Proc. MOBICOM, 2002.

45. Tiny, O.S. A Component-based OS for the Network Sensor Regime. http://webs.cs.berkeley.edu/tos/, October, 2002.

46. Triantafilloy, P.; Ntarmos, N.; Nikoletseas, S.; Spirakis, P. NanoPeer Networks and P2P Worlds. In Proc. 3rd IEEE International Conference on Peer-to-Peer Computing, 2003.

47. Ye, W.; Heidemann, J.; Estrin, D. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In Proc. 12th IEEE International Conference on Computer Networks (INFOCOM), New York, NY, USA, June 23-27 2002.

48. Wireless Integrated Sensor Networks: http:/www.janet.ucla.edu/WINS/, April, 2001.

49. Xu, Y.; Heidemann, J.; Estrin, D. Geography-informed energy conservation for ad-hoc routing. In Proc. ACM/IEEE International Conference on Mobile Computing (MOBICOM), Rome, Italy, July 16-21 2001.

50. Warneke, B.; Last, M.; Liebowitz, B.; Pister, K.S.J. Smart dust: communicating with a cubic-millimeter computer. *Computer* **2001**, *34*, pp.44-51.

51. Zhang, W.; Cao, G. Optimizing tree reconfiguration for mobile target tracking in sensor networks. In Proc. IEEE InfoCom, 2004.

52. Zhao, F.; Guibas, L. *Wireless Sensor Networks: An Information Processing Approach*; Morgan Kaufmann Publishers, San Fransisco, CA, USA; 2004.

53. Zou, Y.; Chakrabarty, K. Sensor deployment and target localization based on virtual forces. In Proc. IEEE InfoCom, San Francisco, CA, April 2003; pp. 1293-1303.

54. Ringwald, M.; Roemer, K. BitMAC: a deterministic, collision-free, and robust MAC protocol for sensor networks. In Proc. of the Second European Workshop on Sensor Networks, 2005; pp. 57-69.

55. Oh, S.; Russell, S.J.; Sastry, S.S. Markov Chain Monte Carlo Data Association For General Multiple-Target Tracking Problems. In Proc. Fourty-third IEEE Conference on Decision and Control; IEEE Press: Piscataway, NJ, USA, 2004; Vol. 1, pp. 735-742.

56. Demirbas, M.; Arora, A.; Gouda, M.G. A Pursuer-Evader Game for Sensor Networks. In Proc. Self-Stabilizing Systems, San Francisco, CA, USA, 2003; pp. 1-16.

57. He, T.; Gu, L.; Luo, L.; Yan, T.; Stankovic, J.A.; Son, S.H. An overview of data aggregation architecture for real-time tracking with sensor networks. In Proc. IPDPS, Rhodes Island, Greece, 29 April 2006.

58. Powell, O.; Leone, P.; Rolim, J. Energy optimal data propagation in wireless sensor networks. *J. Parallel Distrib. Comput.* **2007**, *67*, 302-317.

59. Jarry, A.; Leone, P.; Powell, O.; Rolim, J. An Optimal Data Propagation Algorithm for Maximizing the Lifespan of Sensor Networks. In Proc. DCOSS, San Francisco, CA, USA, June 18-20 2006.