Close-to-optimal Energy Balanced Data Propagation via Limited, Local Network Density Information

Azzedine Boukerche
PARADISE Research Laboratory,
University of Ottawa, Canada
boukerch@site.uottawa.ca

Dionysios Efstathiou
Research Academic Computer Technology Institute (CTI) and University of Patras, Greece
eustathi@ceid.upatras.gr

Sotiris Nikoletseas
Research Academic Computer Technology Institute (CTI) and University of Patras, Greece
nikole@cti.gr

Christoforos Raptopoulos
Research Academic Computer Technology Institute (CTI) and University of Patras, Greece
raptopox@ceid.upatras.gr

ABSTRACT

We study the problem of energy-balanced data propagation in wireless sensor networks. The energy balance property is crucial for maximizing the time the network is functional, by avoiding early energy depletion of a large portion of sensors. We propose a distributed, adaptive data propagation algorithm that exploits limited, local network density information for achieving energy-balance while at the same time minimizing energy dissipation.

We investigate both uniform and heterogeneous sensor placement distributions. By a detailed experimental evaluation and comparison with well-known energy-balanced protocols, we show that our density-based protocol improves energy efficiency significantly while also having better energy balance properties.

Furthermore, we compare the performance of our protocol with a centralized, off-line optimum solution derived by a linear program which maximizes the network lifetime and show that it achieves near-optimal performance for uniform sensor deployments.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: [Wireless Communication]

General Terms

Algorithms, Design, Performance

Keywords

Wireless Sensor Networks, Algorithms, Data Propagation, Energy Balance, Routing, Performance Evaluation

MSWiM'11, October 31–November 4, 2011, Miami, Florida, USA.

1. INTRODUCTION

We study Wireless Sensor Networks (WSNs) envisioned as very large collections of autonomous smart sensor nodes, which distributively form an ad-hoc wireless communication network. The nodes are deployed in a region of interest in order to monitor crucial events and propagate sensed data to a base station usually called the *Sink*. WSNs enable important applications ranging from smart/green building automation to environmental monitoring.

Motivation and Related Work. In WSNs it is crucial to design protocols that are energy efficient and maximize the network lifetime. Towards this goal, diverse approaches including hop-by-hop transmission techniques [5], as well as clustering algorithms [8] and power-saving modes [6] have been proposed.

All the above techniques do not explicitly take care of possible overuse of certain sensors in the network. For example, in a hop-by-hop propagation protocol where data must be forwarded to the *Sink*, the sensors in the area around the *Sink* tend to be overused creating a bottleneck area, since these nodes have to relay the data for other nodes, resulting in an unequal energy consumption. Thus, these nodes may die out very early resulting to network disconnection, although there may be still significant amounts of energy in the other sensors of the network. Also, distant transmissions in cluster head schemes, tend to overuse the cluster heads as the cost of a transmission is roughly some power of the distance between the communicating. The rotation of cluster heads may not be enough (especially in large networks with high event generation rate) to avoid their energy depletion.

Lifespan maximization can be achieved by keeping energy balance [3]. To overcome the unbalanced energy consumption, the authors in [4] proposed a mixed routing strategy where in each step the algorithm decides probabilistically and locally whether to propagate data one-hop towards the *Sink*, or to send it directly to the *Sink*. In [4], it was shown that such a randomized mixed strategy can be used to balance the energy consumption among nodes and increase the lifespan of the network. However, only homogenous topologies and uniform data generation rates are studied, while the energy efficiency itself is not explicitly addressed.

In [2] the authors propose an on-line distributed algorithm

for lifespan maximization where in each step the algorithm decides to propagate data one-hop towards the *Sink* or to send it directly to the *Sink* based however on explicit information about the energy spent by the current node and the energy spent by the nodes that are one-hop closer to the *Sink*. No topology heterogeneity is assumed and energy efficiency is only indirectly taken into account.

A nice, related but different, approach for achieving load balancing is proposed in [9], where the authors propose a non-uniform node distribution deployment to achieve energy-balanced consumption in the network. More precisely, in the (off-line) network deployment phase, they regulate the number of nodes in each sector in order to tune the ratio between the node densities in the adjacent (i+1)-th and i-th sectors. For this deployment, they also present an efficient routing scheme (q-switch) that achieves energy balance using single hop transmissions. Our protocol can be used for a broad class of network deployments and achieves energy balance in a different way, by using a combination of short, long and "bypass" transmissions.

In [1] the authors suggest exploiting mobility to lower transmission power thus indirectly leading to a more uniform energy dissipation.

The Problem. We study the problem of how to achieve energy-balanced data propagation in distributed WSNs for both uniform and heterogeneous sensor deployments. The energy balance property guarantees that the average energy spent per sensor is the same for all sensors in the network at any time during the network operation. This property is crucial for prolonging the network lifetime by avoiding early energy depletion of sensors and the non-utilization of available energy on sensors. Together with energy balance we also want (in contrast to previous work) to optimize the energy efficiency of data propagation.

Our Contribution. Our study was originally motivated by [7] where the authors define the energy balance property and propose an energy-optimal and energy balanced algorithm for the particular problem of sorting in WSNs.

We propose an on-line, distributed algorithm which takes into account limited, local density information towards balancing the energy consumption and thus increasing the lifespan of the network. Apart from the usual well-studied random uniform sensor deployments, we also study, for the first time in the context of the energy balance property, the more realistic case of a heterogeneous sensor placement by introducing a general model of non-uniform deployment. In the relevant state of the art, energy-balancing protocols have not been studied in heterogeneous node placement settings.

The experimental results show that our density-based heuristic actually works well for both uniform and non-uniform network deployments and has much better performance compared to other well-known energy balance data propagation algorithms for both sensor placement distributions and various node densities. In particular, its performance for uniform sensor deployment (as shown via comparing to an off-line optimum evaluated by a linear program) is near-optimal.

2. MODEL

We consider a plane sensor network, in which the sensors and the single *Sink* node are static. We abstract the network by a graph G(V,E), where V denotes the set of nodes (sensors), while $E \subseteq V^2$ represents the set of edges (wireless links). An edge between two nodes in the graph exists iff the distance between the corresponding sensors in the network is below the transmission range R. The distance between nodes is Euclidean, and the path length is the sum of the hops of the intermediate pairs of subsequent nodes.

Without loss of generality, we assume that network deployment area is a circle of radius D. We virtually slice the network into $M = \frac{D}{R}$ co-centric *Rings* and $N = \frac{2\pi}{\phi}$ *Slices*, where $\phi = \frac{\pi}{6}$. A sector is defined as the intersection of a specific ring and slice. For example, in Fig. 1 the network is divided into 12 Slices where each Slice contains 20 Sectors, resulting in a total of 240 sectors in the network.

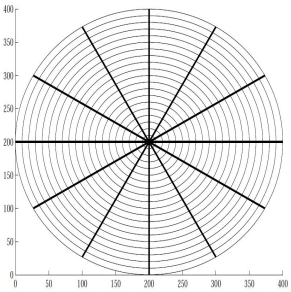

Figure 1: The sectorized network model.

Each node knowns its location, has an unique ID and belongs to exactly one Sector of a Slice. Using this information, each node can identify in which Slice and Sector it belongs to. Nodes are aware of which nodes belong to neighboring Sectors and the position of the static Sink. This information can be disseminated through a set-up phase initiated by the Sink during which the position of the Sink and the IDs of the nodes in neighboring Sectors are diffused.

Uniform deployment. In this case, we deploy the n nodes of the network uniformly at random in the network deployment area, see Fig. 2 (left). Notice that the expected density is the same for all sectors in the network.

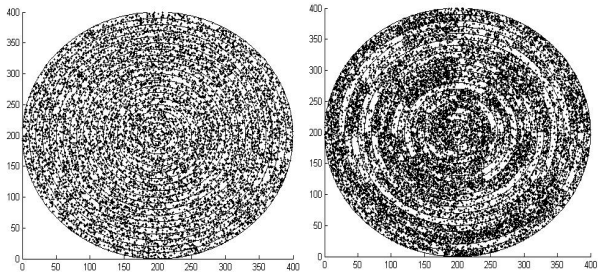

Figure 2: Uniform deployment (left) and Non-uniform deployment (right).

Non-uniform deployment. Our experiments are also performed in random instances of the following general model of non-uniform deployment: Denote by $S_{i,j}$ the sector corresponding to the intersection of slice i and ring j. Let $c > 1$ be an arbitrary constant. Each sector $S_{i,j}$ chooses independently a number $\delta_{i,j} \in [1, c]$ according to the uniform distribution $\mathcal{U}[1, c]$. We will refer to the number $\delta_{i,j}$ as the *relative density* of sector $S_{i,j}$. Clearly, values of $\delta_{i,j}$ close to 1 imply low relative density and values close to c imply high relative density. By combining the knowledge about the total number of sensors in the network n, together with

the relative density $\delta_{i,j}$ and the area $A_{i,j}$ of every sector, we compute the number of nodes $n_{i,j}$ deployed in sector $S_{i,j}$ by the following formula:

$$n_{i,j} = \frac{n}{\sum_{i',j'} \frac{A_{i',j'}}{A_{i,j}} \frac{\delta_{i',j'}}{\delta_{i,j}}}.$$

where $n = \sum_{i,j} n_{i,j}$. Finally, we scatter $n_{i,j}$ nodes in the area corresponding to sector $S_{i,j}$, see Fig. 2 (right).

Notice that the fraction of the actual densities of two sectors $S_{i,j}$ and $S_{i',j'}$ is exactly $\frac{A_{i',j'}}{A_{i,j}} \frac{\delta_{i',j'}}{\delta_{i,j}}$ (which also justifies the name relative density for $\delta_{i,j}$). Furthermore, if all sectors have the same relative density (i.e. $\delta_{i,j} = \delta_{i',j'}$, for all i', j'), we get the uniform deployment.

Our protocols operate at the network layer, so we are assuming appropriate underlying data-link, MAC and physical layers. The nodes' memory is assumed limited e.g. we allow messages to piggy-back a constant number of bits of information only, encoding the position of the last node visited and the position of the *Sink*.

We consider two types of data transmission: a) single-hop transmission (cheap in terms of energy and slow) between two nodes belonging to neighboring sectors and b) direct transmission (expensive in terms of energy and fast) where the node that holds the data transmits directly to the *Sink*.

We assume that the energy spent at a sensor when transmitting data messages is proportional to the square of the transmitting distance. The size of messages is considered to be constant. We only count (for simplicity) energy spent during transmissions.

3. THE OFF-LINE OPTIMAL SOLUTION TO THE PROBLEM

Consider now a specific slice in our sensor network model. This slice contains N sectors, which we denote by $S_1, S_2, \ldots, S_N$, dropping the slice number of the sectors for the sake of simplicity. Every sector S_i contains n_i sensors and each sensor has $E_{initial}$ initial energy at the beginning of the network operation. Given a specific slice, we model the routing of messages with a linear program (LP) (see below definitions 4 - 5).

Definition 1. λ is the number of messages produced by each sensor in the network.

Definition 2. $E_{initial}$ is the initial energy of each node in the network.

Definition 3. n_i is the number of nodes in sector i, where $1 \leq i \leq N$.

Definition 4. The λ maximization problem is on input $\{E_{initial},\ n_i\}$, to find the maximum number of messages that every sensor can produce with given initial energy and node n_i distribution per sector. If we multiply λ with the total number of nodes in the network, we get the maximum number of messages throughout the network that are routed to the *Sink*.

Maximize λ:

$$\sum_{j=1}^{N} f_{ij} \left(i-j\right)^2 \cdot R^2 \leq n_i * E_{initial}, \qquad 1 \leq i \leq N \qquad (1)$$

$$\sum_{j=0}^{N} f_{ij} = \sum_{j=1}^{N} f_{ji} + \lambda \cdot n_i, \qquad 1 \leq i \leq N \qquad (2)$$

$$f_{ij} = 0, \qquad j \notin \{0,\ i-1\}, \qquad 1 \leq i \leq N \qquad (3)$$

$$f_{ij} > 0, \qquad j \in \{0,\ i-1\}, \qquad 1 \leq i \leq N \qquad (4)$$

In Equation 1, f_{ij} is the number of messages routed from sector i to sector j while not spending more energy than available. Eq. 2 guarantees flow preservation i.e. all messages generated at or forwarded to a sector are pushed out of it. Eq. 3 and Eq. 4 allow only next-slice transmissions and direct transmissions to the sink. Equations 5 to 8 are similar. Notice that in the above linear program, λ is a variable of the LP and $E_{initial}$ is given as input, while in the following linear program $E_{initial}$ is a variable of the LP and λ is predefined.

Definition 5. The $E_{initial}$ minimization problem is on input $\{\lambda,\ n_i\}$, to find the minimum initial energy $E_{initial}$ which achieves the desired λ data generation rate per node in a given n_i node distribution. Thus, by multiplying the output $E_{initial}$ of the LP with the number of nodes we get the optimal energy consumption of the whole network.

Minimize $\mathbf{E_{initial}}$:

$$\sum_{j=1}^{N} f_{ij} \left(i-j\right)^2 \cdot R^2 \leq n_i * E_{initial}, \qquad 1 \leq i \leq N \qquad (5)$$

$$\sum_{j=0}^{N} f_{ij} = \sum_{j=1}^{N} f_{ji} + \lambda \cdot n_i, \qquad 1 \leq i \leq N \qquad (6)$$

$$f_{ij} = 0, \qquad j \notin \{0,\ i-1\}, \qquad 1 \leq i \leq N \qquad (7)$$

$$f_{ij} > 0, \qquad j \in \{0,\ i-1\}, \qquad 1 \leq i \leq N \qquad (8)$$

4. THE PROTOCOLS

4.1 The P_i Energy Balance Protocol

The authors in [4] propose an energy balance algorithm where in each step the node that holds the data decides whether to propagate data one-hop towards the *Sink* with probability p_i , or to send data directly to the *Sink* with probability 1 - p_i, where i is the current sector. The appropriate probability p_i is calculated as follows:

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

where x is a free (e.g. controlling) parameter and i is related to the sensor'Ss distance to the *Sink*. This randomized choice balances the (cheap but slow) one-hop transmissions with the (expensive but fast) direct transmissions to the *Sink*, which are more expensive but "bypass"the nodes lying close to the *Sink*. Note that, in most data propagation protocols, the bottleneck area around the *Sink* tends to

be overused and die out early. The lifespan of the network is maximized by ensuring that the energy consumption in each slice is the same. Sensors are assumed to be randomly distributed with uniform distribution.

4.2 The E_i Energy Balance Protocol

In [2] the authors propose an on-line distributed algorithm for lifespan maximization (via energy balance) of a WSN in a data propagation scenario. Let n be a sensor. V_n is the neighborhood of node n. Node n knows the energy spent by each node in its neighborhood. Let m be the sensor of neighborhood V_n with the lowest energy spent. When n holds data it makes the following decision:

- If node n has spent more energy than m, then n sends the message to m (spending one energy unit).
- Otherwise, n sends the message directly to the *Sink* spending d^2 energy units, where d is the distance from n to the *Sink*.

4.3 Our Density-Based Protocol

The basic idea of our protocol is to take into account information about the density of the Previous and the Next Sectors in the current Slice in order to take a decision to forward data to the next Sector or directly to the *Sink*.

Definition 6. Let $d_{i,j}$ be the density of the j-th Sector in the i-th Slice.

Definition 7. If S is the current Slice and r is the current Sector of the node, let $F_{\frac{P}{N}}$ be the fraction of the density of the Previous β Sectors over the density of the Next β Sectors, which is calculated as follows:

$$F_{\frac{P}{N}} = \frac{\sum_{j=r+1}^{r+1+\beta} d_{S,j}}{\sum_{j=r-1-\beta}^{r-1} d_{S,j}}$$

We provide three options in order to examine the ratio of the densities of the Previous Sectors to the densities of the Next Sectors:

a) **Local Knowledge** ($\beta = 1$). Firstly, to take into account the density of the Previous Sector and the Next Sector only. If S is the current Slice and r is the current Sector, we compute the density ratio as follows:

$$F_{\frac{P}{N}} = \frac{d_{S,r+1}}{d_{S,r-1}}$$

b) **Partial Knowledge** ($\beta = 3$). Secondly, to take into account the density of the Previous β Sectors and the Next β Sectors; β is taken 3 as a good compromise of information and cost. If S is the current Slice and r is the current Sector, we compute the density ratio as follows:

$$F_{\frac{P}{N}} = \frac{\sum_{j=r+1}^{r+1+\beta} d_{S,j}}{\sum_{j=r-1-\beta}^{r-1} d_{S,j}}$$

c) **Global Knowledge** ($\beta = N$). Thirdly, to take into account the density of the all Previous Sectors and the all Next Sectors. If S is the current Slice, r is the current Sector and $T_{Sectors}$ is the total number of Sectors, we compute the density ratio as follows:

$$F_{\frac{P}{N}} = \frac{\sum_{j=r+1}^{T_{Sectors}} d_{S,j}}{\sum_{j=1}^{r-1} d_{S,j}}$$

We characterize the density of the Next or Previous Sectors as "high", when the average density of the Next or Previous β Sectors is M times larger than the global network density D_{Global}; in particular we chose M to be equal with 3. Similarly, we characterize the density of the Next or Previous Sectors as "low", when the average density of the Next or Previous β Sectors is M times lower than the global network density D_{Global}.

Definition 8. If S is the current Slice and r is the current Sector of the node, let $F_{Previous}$ be the fraction of the density of the Previous β Sectors over the global network density, which is calculated as follows:

$$F_{Previous} = \frac{\sum_{j=r+1}^{r+1+\beta} d_{S,j}}{D_{Global}}$$

Definition 9. If S is the current Slice and r is the current Sector of the node, let F_{Next} be the fraction of the density of the Next β Sectors over the global network density, which is calculated as follows:

$$F_{Next} = \frac{\sum_{j=r-1-\beta}^{r-1} d_{S,j}}{D_{Global}}$$

4.3.1 Our Data Propagation Scheme

The node that holds the data to be delivered to the *Sink* decides to act suitably according to the relation of densities in previous and next sectors towards the sink.

a) $\mathbf{F_{\frac{P}{N}} < (1 - \alpha)}$, where α constant such that $0 < \alpha < 1$. The particular value of constant α can be chosen by the protocol designer to fine-tune performance: a large α value would lead to less energy consumption but a higher data delivery latency while a smaller α would incur a smaller data delivery latency but it would increase a the energy consumption. The experimental evaluation suggested that the value $\alpha = 0.7$ leads to an optimized trade-off. If the density of the Next β Sectors is much higher than the density of the Previous β Sectors, the node forwards data as following:

- If both the Next and the Previous β Sectors have high densities then the node forwards data hop-by-hop, since the forward area can afford the load of the previous area.
- If both the Next and the Previous β Sectors have low densities then the node forwards data using a probability p_1. The probability p_1 randomizes the propagation of the message one-hop towards the *Sink* or directly to the *Sink* as follows:

$$\text{Probability} = \begin{cases} \gamma_1, & \text{"Jump" to the Sink} \\ 1 - \gamma_1, & \text{Hop-by-hop transmission} \end{cases}$$

In fact, a small, constant probability γ_1 where $\gamma_1 > 0$ for jumps significantly improves performance, while larger values are not appropriate due to the high energy cost of jumps. The value of $\gamma_1 = \frac{1}{30}$ has been chosen after many simulations tries suggesting that

this value optimizes performance. Again, the protocol designer can freely choose γ_1 achieve particular tradeoffs: a large γ_1 would lead to faster but more expensive data propagation while a small one would result to more conserving energy consumption but slower data propagation. This selection encourages a lot the hop-by-hop transmissions. The jump transmissions are expensive for the sender node but do not overload the front network area.

b) $(1-\alpha) < \mathbf{F}_{\frac{\mathbf{P}}{\mathbf{N}}} < (\mathbf{1}+\alpha)$, where α a constant as in case a). This suggests a threshold behaviour for $\mathbf{F}_{\frac{\mathbf{P}}{\mathbf{N}}}$ around the relative density value 1. This means that if the density of the Next β Sectors and the Previous β Sectors are approximately the same, the node should forward data as follows:

- If the Previous and the Next β Sectors have high densities, comparing with the Global density then the node forwards data using a probability p_2 which randomizes the propagation of the message one-hop towards the *Sink* or by-passing the overloaded front area as follows:

$$\text{Probability} = \begin{cases} \gamma_2, & \text{By-pass transmission} \\ 1-\gamma_2, & \text{Hop-by-hop transmission} \end{cases}$$

  As in the case a) for jumps, a similar behavior applies this case for bypasses: a very small constant bypass probability γ_2, where $\gamma_2 > 0$ a very small constant suffices to improve performance a lot, with the protocol implementor being free to select the particular value accordingly. In by-pass transmission a node chooses randomly between the left and the right diagonial sector towards the *Sink* in order to transmit its data. The value of probability $1-\gamma_2$ has been chosen to be $\frac{29}{30}$ (i.e. $\gamma_2 = \frac{1}{30}$) after fine-tuning, encouraging a lot the hop-by-hop transmissions, as the front area is dense enough to transmit the data from the backward network area. However, the by-pass transmissions are used in order to avoid the overloaded front network area.
- If the Previous and the Next β Sector have low densities, the node forwards data to the next Sector, because the front area can relay the load of the backward network area.

c) $\mathbf{F}_{\frac{\mathbf{P}}{\mathbf{N}}} > (\mathbf{1}+\alpha)$, where $0 < \alpha < 1$ a constant as in cases a) and b). If the density of the Next β Sectors is smaller than the density of the Previous β Sectors, the node forwards data using a probability p_3 which randomizes the propagation of the message one-hop towards the *Sink* or by-passing the overloaded front area as follows:

$$\text{Probability} = \begin{cases} \gamma_3, & \text{By-pass transmission} \\ 1-\gamma_3, & \text{Hop-by-hop transmission} \end{cases}$$

Similarly to cases a) and b), here again a very small, arbitrarily chosen bypass probability γ_3, where $\gamma_3 > 0$ a small constant is used. The particular value of probability $\gamma_3 = \frac{19}{20}$ (i.e. $\gamma_3 = \frac{1}{20}$) has been chosen after many simulations tries suggesting that this value optimizes performance. This value of probability γ_3 encourages a lot the the hop by hop transmissions over the by-pass transmissions. The by-pass transmissions are used in order to avoid the overloaded sparse front network area.

Overall, this algorithmic design is very simple basically suggesting a threshold behaviour around the value 1 for the relation of "forward" and "backward" network densitites. Also, a probabilistic, adaptive addition of a small amount of jumps or bypass transmissions (on top of multi-hop data propagation, which is the prevailing propagation method) is able to incur significant performance gains.

5. EXPERIMENTAL EVALUATION

Experimental Setup. Our simulation environment for making the experiments is Matlab 7.9.0. We consider two types of node deployment: a) **random uniform** and b) **heterogenous**. The transmission range R is taken 10, the initial energy in each sensor is $E_{initial} = 10^6$ and we conducted experiments for various node densities ($10{\cdot}10^3$, $15{\cdot}10^3$, $20{\cdot}10^3$ and $25{\cdot}10^3$ nodes). As depicted in Fig. 1, the circle network of radius 200 is divided into 12 Slices where each Slice contains 20 Sectors.

In detail, the network area is rectangular, with length and width equal to 40 units. For statistical smoothness, we apply several times the deployment of nodes in the network and repeat each experiment 50 times. For each experiment we simulate 30.000 data propagations and the average value is taken. The statistical analysis of the findings (the median, lower and upper quartiles, outliers of the samples) demonstrates very high concentration around the mean, so in the following figures we only depict average values.

The *Sink* is placed at the center $(100, 100)$ of the circle deployment area. In simulation experiments, we focus on the following metrics: a) **data delivery latency** which is the average number of hops needed to reach the *Sink*; b) **average energy consumption** is the average energy spent per node during the network operation; c) **variation of energy consumption** which is the deviation of the energy consumed by a node from the average energy consumption in the network. This metric is useful to determine whether the energy consumption in a network is energy-balanced, e.g. as it is impossible to achieve a close to zero deviation, the lower is the deviation the more energy-balanced is the network and vice versa; and d) **success ratio** is the ratio of the total number of packets received by the sink to the number of packets generated by the whole network.

We assume an energy model, in which the energy consumed by a message transmission between two nodes is considered the square of the distance between the nodes. For the above metrics the average is taken over all sensor deployments and algorithm's repetitions.

5.1 The impact of β (network information)

In order to choose the ideal β we have conducted a detailed experimental comparison of our protocol performance for various choices of β. As described previously, β is the amount of knowledge required by our protocol. More specifically, β is the number of the Previous β and Next β Sectors of which a node needs to know their density. In Section 4.3.1, we proposed the following basic choices about β, a) local knowledge of the network ($\beta = 1$) in which a node

knows the density of the neighboring Sectors only, b) partial knowledge ($\beta = 3$) where a node knows the densities of the three Previous and Next Sectors and c) global density information ($\beta = N$) where a node has global information about the densities of all the N Sectors in the Slice.

We concluded that the best choice of β is 3, as it is feasible to get distributively Partial knowledge of the neighboring Sectors and has better performance from the case of local knowledge $\beta = 1$.

5.2 Comparison of Protocols

For the uniform deployment scenario, Fig. 3 (top) shows a negative impact of our Density-Based Protocol on data delivery latency and its performance is close to P_i's results. On the other hand, the E_i Protocol is the fastest of all as it makes a lot of direct transmissions. As the density of the network increases, the E_i Protocol becomes faster while the performance of the other two protocols remains stable.

The energy-latency trade-off is more clear if you observe the Fig. 3 (middle) that illustrates the average energy consumption per node. In more detail, the Density-Based Protocol is the most energy conservative. As expected the E_i Protocol depletes the energy of the nodes quickly and its energy-efficiency gets worse as the number of the network nodes increases.

Fig. 3 (bottom) depicts the variation of energy consumption which is a metric of the energy-balance property. Our Density-based Protocol has at least 4 times lower deviation from the average energy consumption than the deviation of the other two protocols. Our protocol's behaviour improves remarkably as the network density increases. We can claim that our Density-Based Protocol prolongs the lifetime of the network more than the other two energy-balance protocols. This result indicated that knowledge of the densities of neighboring areas (or more precisely, predicting the traffic forwarded to and received from other network areas) can lead to energy balance consumption. The P_i Protocol has the worst energy-balance behavior and this can be explained by the fact that it uses only proximity information in order to decide the next-hop transmission.

For the non-uniform deployment, the protocols have the same behavior for the data latency metric as shown in Fig. 4 (top). The results for the average energy consumption metric are shown in Fig. 4 (middle). The E_i Protocol is more expensive in the case of heterogeneous deployments rather than the uniform and the most expensive of all the protocols for both deployments. Also, in the heterogeneous case, the E_i Protocol has lower variation of energy consumption, see Fig. 4 (bottom), and for smaller network densities ($10*10^3$, $15*10^3$ nodes) has even better energy-balance performance than our Density-based Protocol. However, our Density-based Protocol's energy-balance behavior improves as the network density increases and achieves a more energy-efficient balanced consumption than the E_i protocol which has similar energy-balance behavior but in a less efficient way as it spends 3 times more energy than the Density-based Protocol.

5.3 Energy-Balance Property

In this section we study the energy-balance property of the protocols. In detail, we present graphically the spatial evolution of energy dissipation in a network of 20.000 nodes after 30.000 data propagations for both random uniform (see

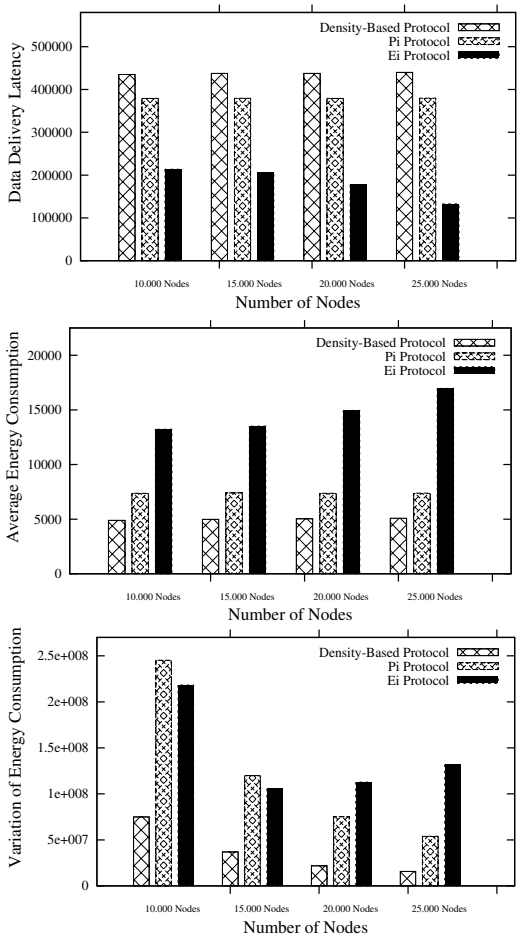

Figure 3: Experimental Comparison of the four protocols for various node densities in Uniform Network Deployment.

Fig. 5) and heterogeneous (see Fig. 6) sensor deployments. Nodes with high energy dissipation are depicted with dark colours. In contrast, nodes with high residual energy are depicted with bright colours.

We observe in Fig. 5 (left), that our Density-based Protocol achieves better energy balancing than the P_i Protocol Fig. 5 (center) and the E_i Protocol Fig. 5 (right). In detail, P_i Protocol tends to overuse the nodes that are in the middle in each sector, which can be explained by the random choice of the node of the next Sector as it is more probable to choose a node which is in the middle of the Sector area. The E_i Protocol exhausts the distant nodes as it makes many expensive jumps at the start of the network operation. On the other hand, our Density-based Protocol spends the energy of the sensors in a more balanced way.

Interestingly, in the scenario of heterogeneous node deployment depicted in Fig. 6 our Density-based protocol Fig. 6 (left) is again more energy-balanced than the P_i Protocol Fig. 6 (center) and the E_i Protocol Fig. 6 (right).

5.4 Comparing with the optimum

In this section, we compare the performance of the three energy-balance protocols (Density-based, P_i and E_i Proto-

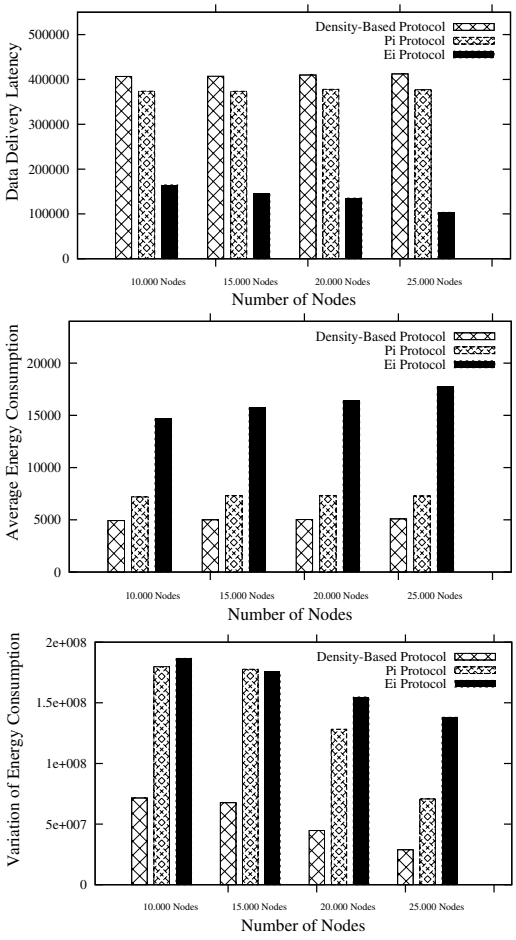

Figure 4: Experimental Comparison of the four protocols for various node densities in Non-Uniform Network Deployment.

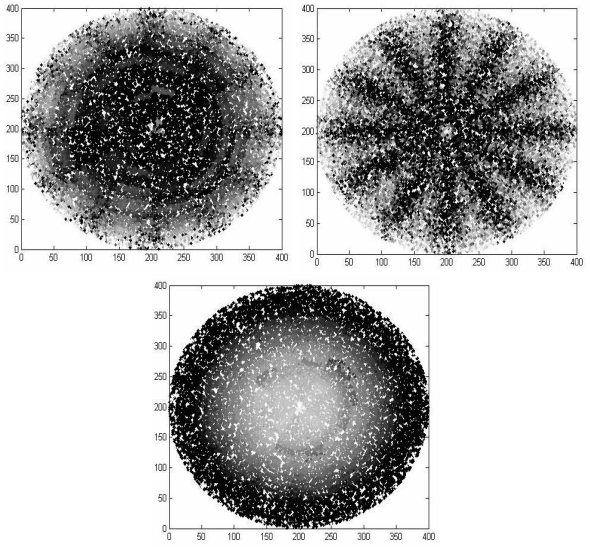

Figure 5: Energy dissipation map in uniform deployment using the Density-Based (top left), P_i (top right) and E_i Protocol (bottom).

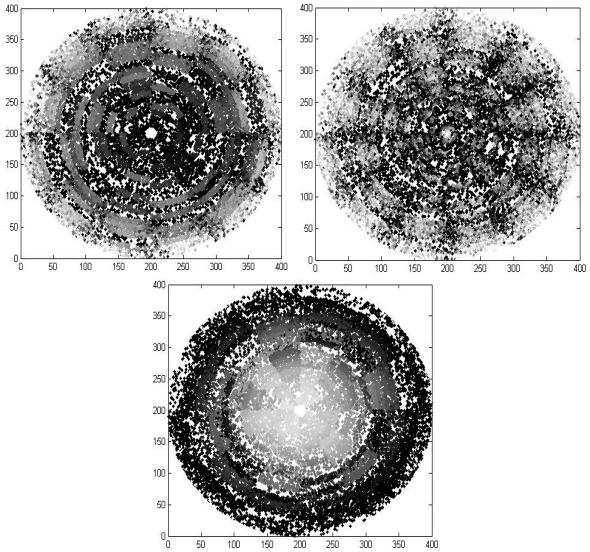

Figure 6: Energy dissipation map in non-uniform deployment using the Density-Based (top left), P_i (top right) and E_i Protocol (bottom).

col) with the centralized off-line optimal solution as given by the linear program described in Definition 5.

In the case of uniform deployment, Fig. 7 shows that the success ratio of Density-based Protocol is consistently higher during the network operation as the number of messages generated per node increases. The success ratio of E_i is close to our protocol's performance. On the other hand, P_i has the worst performance in terms of success ratio because it has the worse energy-balance behavior, and the nodes close the critical area around the *Sink* are depleted very early.

The energy consumption results are shown in Fig. 8. As expected, the E_i Protocol is the most expensive as it makes a lot of jumps at the start of the network operation. Our protocol is the most energy efficient of all and has close to optimal performance in terms of energy consumption. The P_i Protocol is the second expensive and it is more consuming if you take into consideration that it has very low success ratio as the network continues to operate (for $\lambda > 5$).

In the case of heterogeneous deployment, we remark that the energy consumption of the optimum solution is greatly reduced (Fig. 10), whereas the energy dissipation of other protocols is increased. As noticed in [9], you can achieve the optimum energy-balanced energy consumption in a sensor network by placing the nodes in a heterogeneous manner. If you place the nodes uniformly in the area of interest, the nodes nearer to the sink carry heavier traffic loads and they will deplete their energy faster. For that reason, the optimum for the non-uniform deployment case has better performance from the optimum for the uniform deployment.

6. CONCLUSIONS/FUTURE WORK

Our approach is one of the few first that investigate the problem of maximizing the network lifetime (via energy balance) in the context of heterogeneous sensor deployment. We propose a new distributed, local and lightweight protocol that achieves improved energy balanced data propagation

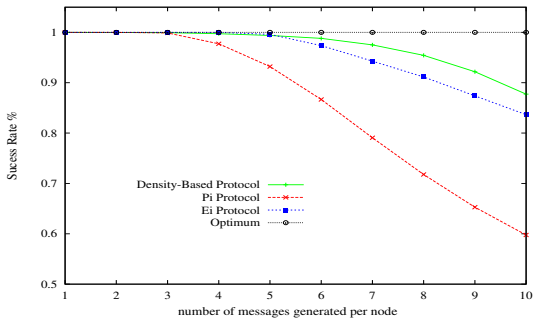

Figure 7: Success rate of the protocols and the optimum in Uniform Network Deployment.

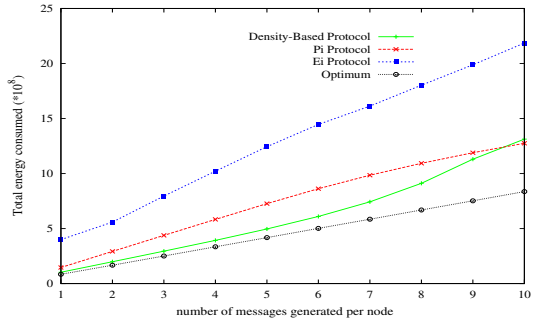

Figure 8: Energy consumption of the protocols and the optimum in Uniform Network Deployment.

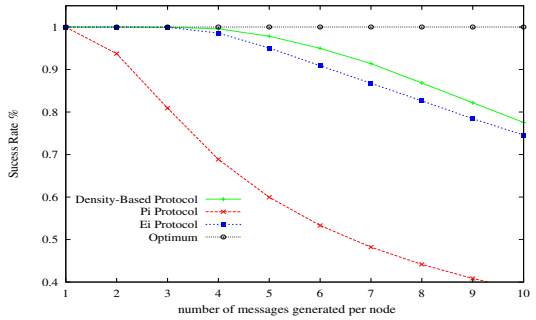

Figure 9: Success rate of the protocols and the optimum in Non-Uniform Network Deployment.

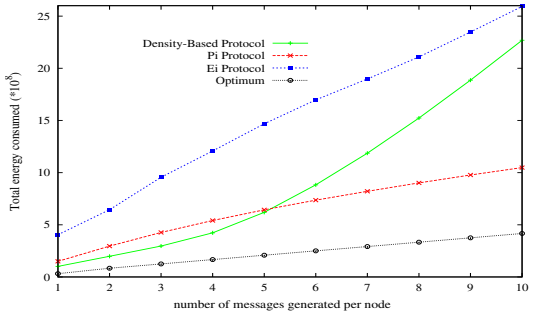

Figure 10: Energy consumption of the protocols and the optimum in Non-Uniform Network Deployment.

than other well known energy balance schemes; the main idea of our method is to exploit some limited knowledge of node density in the neighborhood of the node currently holding data under propagation. As shown by detailed simulation results, in uniform deployment our protocol has near-optimal performance (very close to the centralized off-line optimum solution by a linear program); in particular, our method is energy balanced and energy efficient.

In future work, we intend to study the problem of energy balanced data propagation in sensor networks with mobile nodes. Also, we plan to come up with a mathematically strict way of calculating the probabilities p_1, p_2 and p_3 in the Data Propagation Scheme of our Density-Based Protocol. Furthermore, we plan to implement the above mentioned energy-balanced protocols in a experimental test-bed consisting of Xbow's TelosB motes, and study their performance in real world conditions. We plan to also study the behaviour of our method in the regulated deployment of [9]. Finally, we plan to study the performance of our protocol in various propagation models where the power fall more rapidly than the square of the distance (e.g. $n = 3, n = 4$).

Acknowledgements This work has been partially supported by the EU Project HOBNET (ICT/FIRE STREP 257466).

7. REFERENCES

[1] F. Pasquale A. Clementi and R. Silvestri. Manets: High mobility can make up for low transmission power. In *ICALP*, pages 387–398, 2009.

[2] O. Powell A. Jarry, P. Leone and J. Rolim. An optimal data propagation algorithm for maximizing the lifespan of sensor networks. In *DCOSS*, volume 4026/2006, pages 405–421, 2006.

[3] S. Nikoletseas A. Jarry, P. Leone and J. Rolim. Optimal data gathering paths and energy balance mechanisms in wireless networks. In *DCOSS*, 2010.

[4] S. Nikoletseas C. Efthymiou and J. Rolim. Energy balanced data propagation in wireless sensor networks. *Wireless Networks*, 12(6):691–707, November 2006.

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

[6] S. Ganeriwal C. Schurgers, V. Tsiatsis and M. Srivastava. Topology management for sensor networks: Exploiting latency and density. In *MOBICOM*, pages 135–145, 2002.

[7] M. Singh and V. Prasanna. Energy-optimal and energy-balanced sorting in a single-hop wireless sensor network. In *Proc. First IEEE International Conference on Pervasive Computing and Comminications PERCOM*, 2003.

[8] A. Chandrakasan W. R. Heinzelman and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. 33rd Hawaii International Conference on System Sciences HICSS*, 2000.

[9] G. Chen X. Wu and S. K. Das. Avoiding energy holes in wireless sensor networks with nonuniform node distribution. *IEEE Trans. Parallel Distrib. Syst.*, 19(5):710–720, 2008.