# Avoiding Low-Energy Areas in Wireless Sensor Network Data Dissemination*

**Olga Goussevskaia[1] , Max do V. Machado[1] , Raquel A.F. Mini[2] , Antonio A.F. Loureiro[1] , Geraldo R. Mateus[1] , José Marcos Nogueira[1†]**

[1]Department of Computer Science – Federal University of Minas Gerais
Belo Horizonte, MG, 31270-010, Brazil

[2]Department of Computer Science – Pontifical Catholic University of Minas Gerais
Belo Horizonte, MG, 30535-610, Brazil

{maxm,olga,raquel,loureiro,mateus,jmarcos}@dcc.ufmg.br

***Abstract.*** *One of the most important resources in wireless sensor networks is energy, since, in general, batteries cannot be recharged. The information about the amount of energy available at each part of the network is called the energy map and can be explored by data dissemination algorithms. In this work, a new data dissemination algorithm for wireless sensor networks is proposed. The key idea is to combine concepts presented in trajectory based forwarding with the information provided by the energy map to determine routes in a dynamic fashion. Simulation results revealed that the energy spent with data dissemination activity can be concentrated on nodes with high energy reserves. In this manner, partitions of the network due to nodes that ran out of energy can be significantly delayed and the network lifetime extended.*

***Resumo.*** *Um dos recursos mais importantes em redes de sensores ï¿½a energia, porque, em geral, as baterias nï¿½ podem ser recarregadas. A informaï¿½o sobre a quantidade de energia disponï¿½el em cada parte da rede ï¿½chamada de mapa de energia e esta pode ser explorada por algoritmos de disseminaï¿½o de dados. Neste trabalho, um novo algoritmo de disseminaï¿½o de dados para redes de sensores ï¿½proposto. A idï¿½a principal ï¿½combinar os conceitos presentes na propagaï¿½o baseada em trajetï¿½ia com as informaï¿½es providas pelo mapa de energia para determinar rotas de forma dinï¿½ica. Resultados de simulaï¿½o mostram que a energia nas atividades de disseminaï¿½o de dados pode ser concentrada nos nï¿½ com maior reserva de energia. Dessa forma, partiï¿½es da rede devido a nï¿½ que ficam sem energia podem ser significativamente atrasadas e, com isso, o tempo de vida da rede estendido.*

## 1. Introduction

Wireless sensor networks (WSNs) pose new research challenges related to the design of algorithms, network protocols, and software that will enable the development of applications based on sensor devices [Estrin et al., 1999]. In WSNs, the energy expenditure in data communication is much more compared to data processing. Thus, any communication solution to this kind of network must be power efficient to extend its lifetime.

*This work has been partially supported by the Brazilian agencies CNPq/MCT and CAPES/MEC

†In sabbatical period at universities of Evry and UPMC/Paris6/LIP6, France.

In WSNs, data communication, from the point of view of the communicating entities, can be divided into three cases: from sensor nodes to a sink node, among neighbor nodes, and from a sink node to sensor nodes. Data communication from sensor nodes to a sink node is used to send the sensed data collected by the sensors to a monitoring application. Data communication among neighbor nodes often happens when some kind of cooperation among nodes is needed. Data communication from a sink node to a set of sensor nodes is often used to disseminate a piece of information that is important to those nodes. This kind of data communication is often called data dissemination. Reliable data dissemination is crucial to wireless sensor network. Based on an efficient dissemination algorithm, a sink node can perform different activities, such as to change the operational mode of part or the entire WSN, broadcast a new interest to the network, activate/deactivate one or more sensor nodes, and send queries to the network. In this work, we study the problem of energy-efficient data dissemination.

In [Niculescu and Nath, 2003], the authors propose *Trajectory Based Forwarding* (TBF), a data dissemination technique in which packets are disseminated from a sink to a set of nodes along a predefined curve. The innovation of this approach comes from the definition and treatment of route paths as a continuous function as opposed to a discrete set of points. The key idea is to embed a trajectory in the packet and then let the intermediate nodes forward the packets to those nodes that lie close to the trajectory. Since a trajectory does not explicitly encode the nodes in the path, it is to a large extent impervious to changes in specific nodes that form the topology. Two main advantages of TBF are compact representation, since curves can be described using few parameters, and node independence, since no particular node address is specified in the trajectory.

In this work, an energy-efficient data dissemination algorithm for WSNs, called *Trajectory and Energy-based Data Dissemination* (TEDD), is proposed. The key idea is to combine concepts presented in TBF with the information provided by the energy map[1] [Mini et al., 2005] to determine routes in a dynamic fashion. TEDD is comprised of two main components. The first part is an algorithm for generating trajectories that pass through regions with higher energy reserves and avoid low energy nodes. The main idea is to select a set of nodes in the network that are most suitable for forwarding the packets sent by the sink and to find the best set of curves passing through or near these selected points. The second component of TEDD is a packet forwarding mechanism. The proposed forwarding algorithm is a receiver-based approach, as opposed to a sender-based approach, such as TBF. This characteristic introduces several improvements to the forwarding process. Firstly, it eliminates the need of neighbor table maintenance, which is very expensive in terms of radio transmissions. Moreover, this new forwarding policy presents a more robust behavior in dynamic topology scenarios, such as WSNs.

The rest of this work is organized as follows. Section 2 presents the related work. The process of generating trajectories is showed in Section 3. In Section 4, we describe the new forwarding policy introduced by TEDD. In Section 5, we present simulation results for the scenario where the sink performs several data disseminations to the entire network and to a specific target area. Finally, in Section 6, we present our conclusions and future directions.

## 2. Related work

Several different routing protocols for WSNs have been proposed [Akyildiz et al., 2002]. Due to the nature of WSNs, the basic requirements for routing techniques are scalability and robustness for data dissemination [Ganesan et al., 2001]. The algorithms for these networks have to be designed aiming to extend the network lifetime and, therefore, have to provide both robust communication mechanisms and efficient energy spending.

[1] Energy map is the information about the amount of energy available at each part of the network.

Among all algorithms already proposed in the literature, the most similar to the one proposed in the present work is the Trajectory Based Forwarding (TBF) [Niculescu and Nath, 2003]. TBF is a data dissemination algorithm that uses curve equations to route messages. The trajectory is embedded in each packet and the intermediate nodes make the forwarding decisions based on the trajectory and a neighbor table. To update the neighbor tables, nodes exchange beacon packets periodically. The innovation of this approach comes from the definition and treatment of route paths as a continuous function as opposed to a discrete set of points. Two main advantages of TBF are compact representation (since curves can be described using few parameters) and node independence (since no particular node address is specified in the trajectory).

Figure 1 illustrates the basic operation of TBF. When a node receives a beacon packet, it updates its neighbor table (point *B*). If the received packet is not a beacon, but a data packet, this node checks if it is the node elected to forward the received data packet (point *C*). If it is not true, the node drops the packet (point *D*). If it is the elected forwarding node, it chooses the next hop (point *E*). This choice is made based on its neighbor table and a predefined forwarding policy, e.g., the nearest neighbor to the destination or the nearest neighbor to the curve. After choosing the next hop, the node transmits the packet to the elected neighbor (point *F*).

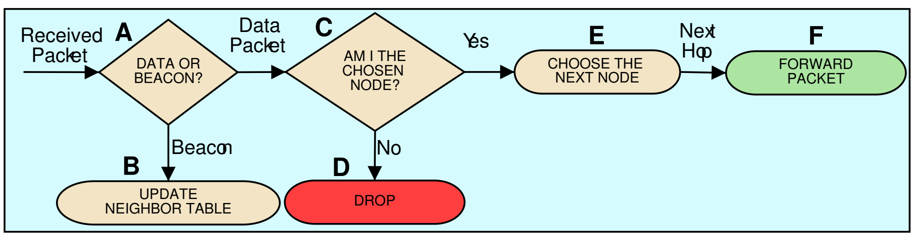



**Figura 1: Basic Operation of TBF.**

Despite the advantages of TBF, this algorithm has three main drawbacks. The first one is the overhead required to update the neighbor tables. It increases the number of transmitted packets and, consequently, increases the total energy spent. The second disadvantage is its weak fault tolerance. This is due to the fact that the next node in a route is determined based on the neighbor table of the previously elected node. Since the decision of forwarding a packet is not made by the node itself, it can be in sleeping mode at the moment when the packet is transmitted to it. The last weakness is due to the fact that each packet embeds exactly one curve, since packets are relayed in a unicast manner. Therefore, a node can select only one neighbor to continue the process and, consequently, only one curve, which may not be sufficient to perform data dissemination to a significant part of the network.

## 3. Dynamic Trajectory Generation

Most of the issues related to determining and specifying the trajectory for TBF are still research topics. In [Niculescu and Nath, 2003], Niculescu and Nath suggest a number of choices for representing a trajectory: *Functional*, *Equational*,*Complex trajectories*, or *Recursive representation*.

In this work, we propose a method for specifying the trajectories dynamically based on the energy map. The main idea is to select a set of nodes in the network that are most suitable for forwarding the packets sent by the sink and to find the best set of curves passing through or near these selected points. The choice of the best set of curves can be based on different criteria, such as the amount of energy available at the forwarding nodes, the percentage of nodes the information disseminated by the sink is supposed to reach, or the area at which the dissemination is aimed.

Since we intend to base the trajectory generation on the energy map, we decided to use functional and equational representations, allowing more than one trajectory to be encoded at each forwarded packet. We consider these two types of representation expressive enough to guarantee the flexibility required to forward packets through areas of greater energy reserves. It is worth noting that low energy areas can change along time due to localized event occurrences or other factors causing irregular energy distribution over the network. The process of generating a set of trajectories is illustrated in Figure 2, and is described throughout the rest of this section.

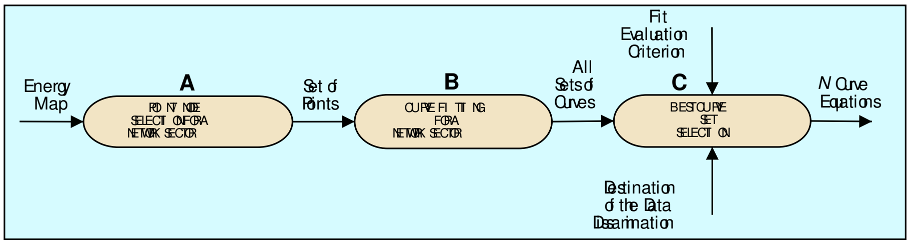



**Figura 2: Trajectory generation process.**

### 3.1. Point/Node Selection

The trajectory generation process requires as input the energy map of the network. The energy map, as described in [Mini et al., 2005], is comprised by geographic coordinates of all nodes, together with the amount of energy available at each node.

The first step in the trajectory generation process is the *point or node selection*, as shown in Figure 2, point *A*. In order to perform the curve fitting process, a set of points has to be selected from the area covered by the WSN to serve as input data for the fitting process. Several strategies can be used to select a subset of the total number of points to serve as input for the curve fitting procedure. The main criterion for this selection is the energy available at each of these points. The idea is to force the trajectories to pass through points with greater energy reserves in order to avoid that nodes with little energy participate in the forwarding process. Another criterion can be the density of node distribution over the network area. The higher is this density, the higher is the network connectivity and, therefore, the probability of packet delivery. This is due to the fact that, since nodes are programmed to periodically go into sleeping mode, there is always a possibility of a trajectory to be broken in case no node is awake in order to receive and transmit the packet.

In this work, we decided to treat the network as a set of nodes, leaving the geographic points perspective for future experiments, and generated the input for the curve fitting procedure by simply selecting 50% of nodes with greater energy reserves.

### 3.2. Curve Fitting

The next step in the trajectory generation process is the *curve fitting*, as shown in Figure 2, point *B*. Due mainly to its simplicity, we decided to use multiple linear regressing [Montgomery et al., 2001] to fit the curves into a set of points. Multiple linear regression attempts to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to the observed data. Formally, given $n$ observations and $p$ estimated parameters, the model for multiple linear regression can be represented as $Y = X\beta + \epsilon$, where $Y$ is an $n$-vector; $X$ is an $n \times p$ matrix; $\beta$ is a $p$-vector with the estimated parameters; and $\epsilon$ is an $n$-vector of random errors. Using the least squares principal, we seek to minimize $(Y - X\beta)'(Y - X\beta)$. Taking the derivative with respect to $\beta$ yields the normal system:

$$(X'X)\beta = X'Y \tag{1}$$

where the estimator $\beta = (X'X)^{-1}X'Y$ follows from solving the normal equations.

Two types of curves were chosen in this work to represent the trajectories: polynomials and conic sections (e.g., ellipses). Polynomials were chosen because of their compact encoding capability for arbitrary network sizes and their flexibility to avoid obstacles or undesired areas. In some scenarios, however, where there are areas of low energy surrounded by areas of high energy, polynomial fitting is not very satisfactory, since it tends to trace the curve through the middle of the low energy area, which is exactly what has to be avoided. Conic sections were chosen because of their better ability to avoid low energy areas in these scenarios.

Polynomial regression is a special case of multiple linear regression and the relation between the response variable $y$ and the explanatory variable $x$ is expressed by the following model:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_g x^g + \epsilon \qquad (2)$$

For this case, simplifying the notation $\sum_{i=1}^{n}$ to $\sum$, and applying (1), the normal system can be written as:

$$\begin{bmatrix} n & \sum x_i & ... & \sum x_i^g \\ \sum x_i & \sum x_i^2 & ... & \sum x_i^{g+1} \\ \sum x_i^2 & \sum x_i^3 & ... & \sum x_i^{g+2} \\ ... & ... & ... & ... \\ \sum x_i^g & \sum x_i^{g+1} & ... & \sum x_i^{2g} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ ... \\ \beta_g \end{bmatrix} = \begin{bmatrix} \sum y_i \\ \sum x_i y_i \\ \sum x_i^2 y_i \\ ... \\ \sum x_i^g y_i \end{bmatrix} \qquad (3)$$

For conic sections, on the other hand, the fitting process is not as direct as for polynomials because they result in a non-linear system. However, if we choose to minimize the squares of the area differences instead of distance differences, we can get a linear problem [Berman, 1983], [M. and Somlo, 1986]. Let us represent a general conic by an implicit second order equation:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \qquad (4)$$

The set of normal equations for this case can be expressed by:

$$\begin{bmatrix} x_1^2 - y_1^2 & x_1y_1 & x_1 & y_1 & 1 \\ x_2^2 - y_2^2 & x_2y_2 & x_1 & y_1 & 1 \\ x_3^2 - y_3^2 & x_3y_3 & x_1 & y_1 & 1 \\ x_4^2 - y_4^2 & x_4y_4 & x_1 & y_1 & 1 \\ x_5^2 - y_5^2 & x_5y_5 & x_1 & y_1 & 1 \\ ... & ... & ... & ... & ... \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} -x_1^2 - y_1^2 \\ -x_2^2 - y_2^2 \\ -x_3^2 - y_3^2 \\ -x_4^2 - y_4^2 \\ -x_5^2 - y_5^2 \\ ... \end{bmatrix} \qquad (5)$$

where $a = 1 + \beta_0$, $b = \beta_1$, $c = 1 - \beta_0$, $d = \beta_2$, $e = \beta_3$ and $f = \beta_4$.

Several iterative methods exist for solving this problem [L. and J., 1974, Paige and Saunders, 1982, R. and E., 1952]. We chose the LSQR algorithm, presented in [Paige and Saunders, 1982], since it presents a more stable behavior and is likely to obtain a more accurate solution in fewer iterations whenever $A$ is moderately or severely ill-conditioned. The computational requirements of this algorithm are: storage $(n + 2p)$, and number of floating-point multiplications per iteration $(3n + 5p)$. In addition, it requires a product $Ax$ and a product $A^t y$ at each iteration. In order to achieve the shown storage, it is required to implement the matrix-vector products in the form $y \leftarrow y + Ax$ and $x \leftarrow x + A^t y$, where $\leftarrow$ means that one of the given vectors is overwritten by the shown expression. The maximum number of iterations was set to $4np$.

### 3.3. Network Sectors

Given a set of points that we would like to force to participate in the forwarding process and given the curve types (polynomial or conic section), we have to decide how many curves/trajectories would be sufficient to achieve a certain goal. The goal could be to disseminate information to a particular area of the network or just perform a broadcast to all nodes.

By introducing the concept of *network sectors*, which divide the network area in identical angular sectors centered at the sink node, the problem of determining the best number of curves/trajectories can be viewed as the problem of finding the best number of network sectors and placing a unique trajectory at each network sector. The curve corresponding to each network sector is fitted based solely on the points located inside that sector. Examples of different sets of network sectors can be viewed in Figures 3-a through 3-d.

An arbitrary number of network sectors could be used. However, it does not seem reasonable this number to be very large, since this would result in an unacceptably high number of parameters to be transmitted with each forwarded packet and an unacceptably low number of points at each sector, compromising the quality of the fitting procedure. A maximum limit can, therefore, be established for the number of network sectors.

### 3.4. Best Curve Set Selection

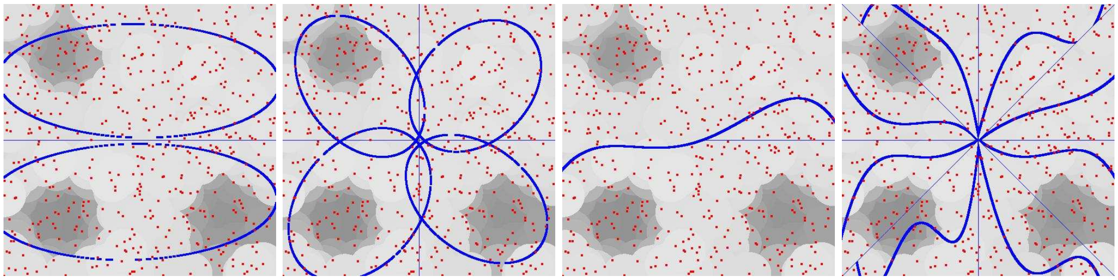
The last step in the trajectory generation process is the *Best Curve Set Selection*, as shown in Figure 2, point *C*. Given a finite maximum number of network sectors, the selection of the best curve set (the best number of network sectors, $N$) can be made by calculating the *average quality* for each set and simply choosing the one with the best average quality. The average quality of a set of curves can be calculated as the sum of the qualities of each curve participating in the set, divided by the number of network sectors in the set. The best number of sectors is determined for each curve type, as shown below:

$$bestNumSec = best\left(\frac{\sum_{i=1}^{numSec} fitQuality(Curve_i)}{numSec}\right), numSec = 1 \ldots maxNumSec \qquad (6)$$

where *maxNumSec* depends on whether the sink node is located in the center or at the corner and whether the curve is polynomial or conic (more network sectors should be tested if the sink node is in the center and the curve is a polynomial). The $fitQuality(Curve_i)$ factor can be calculated based on different criteria, depending on the application requirements. It could be required to forward packets to a maximum number of nodes, or it could be required to minimize the involvement of low energy nodes in the forwarding process. Throughout this work, the following *fit evaluation criteria* were used:

- *Maximum average energy*: calculates the average energy of the nodes within the covering range of the curve ($distance(node, curve) \leq node\_sensing\_range$);
- *Maximum coverage*: calculates the total number of nodes within the covering range of the curve.

In Figure 3, several snapshots of generated curve sets are shown for a given energy map. Figures 3-a and 3-b show the sets of conic sections. It can be observed that the trajectories avoid the low energy areas. Figures 3-c and 3-d show sets of fourth-degree polynomials generated for the same network scenario. It can be observed that once again most of the trajectories avoid the low energy areas. If *maximum average energy* criterion is used to select one of these sets of curves, the set of trajectories using one network sector is chosen. This result is obtained because the average energy of the nodes within the covering range of this curve was higher than the average energy of the rest of the curve sets. If *maximum coverage* criterion is used, on the other hand, the best set is the one that uses eight sectors, since the greater the number of curves, the greater the amount of nodes within their covering range.



(a) Curve set with two network sectors.

(b) Curve set with four network sectors.

(c) Curve set with one network sector.

(d) Curve set with eight network sectors.

**Figura 3: Conic sections and 4th degree polynomial trajectories.**

### 3.5. Restricting the Dissemination Target Area

In the previous discussion, it has been considered that the data dissemination is aimed at the entire network, acting as a broadcast. In some applications, however, the target of the dissemination may be only part of the network, acting as multicast. This situation can arise when the sink node is interested in collecting information from or issuing commands to a specific geographic area.

The curve generation procedure described in this section can be easily adapted to operate under this requirement. Given the coordinates of the boundaries of the dissemination target area, a unique trajectory connecting this area to the sink node can be specified by setting the *maxNumSec* parameter, defined in Section 3.4, to 1, and the *fit evaluation criterion*, also defined in Section 3.4, to *maximum average energy*. In addition, two constrains must be specified for the generated curve: it must intersect the sink node location and intersect the boundary of the target area. In this manner, a polynomial, named *delivery curve*, that avoids low-energy areas will connect the sink to the target area.

Inside the target area, the curve generation procedure can be recursively applied, exactly as described throughout this section. The source of the dissemination, however, is assigned as the intersection point of the connecting polynomial with the target area boundary. Both *maximum average energy* and *maximum coverage fit evaluation criteria* can be used to select the best set of curves inside this area, depending on the application requirements. An example of such a configuration is illustrated in Figure 4. It can be seen that the *delivery curve* avoids the low-energy area located in the middle of the network.

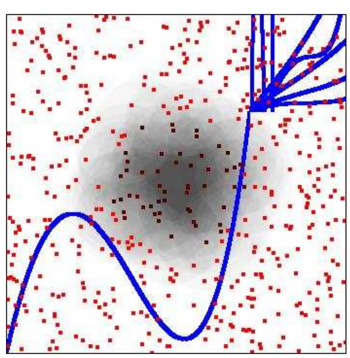


**Figura 4: Dissemination target area (upper right corner) and the delivery curve example.**

### 3.6. Some Remarks

The cost of generating the trajectories is considered irrelevant from the network perspective, since all calculations are made at the sink node, which is considered to have unlimited amount of energy. Therefore, the process of generating the trajectories can be repeated according to application requirements, such as a change in the destination or in the purpose of the dissemination.

It is important to point out that the trajectory generation strategy proposed here is not restricted to the network scenario illustrated in Figure 3. An energy map of a network with an arbitrary shape and an arbitrary number of randomly distributed sink nodes can be used as input to this procedure. In this situation, each node would be able to participate in more than one trajectory, possibly forwarding packets originated by different sink nodes. Another relevant consideration is about the process of encoding the trajectories. Curve parameters can be embedded in the packet header or can be pre-configured in the nodes before delivering them. However, in the latter, the sink node should be able to update those values periodically.

## 4. Data Dissemination Model

In this section, we explain the forwarding process of the TEDD algorithm that tries to overcome some of the drawbacks of TBF. Firstly, TEDD avoids the necessity of neighbor tables, spending less energy in the forwarding process. When a node receives a packet, it decides locally whether it should forward it or not, based solely on the information contained in the packet. Therefore, TEDD does not choose a particular node to forward a packet, the decision of forwarding a packet is made by the node itself. Secondly, TEDD allows to embed more than one curve in each packet, allowing data dissemination to arbitrarily large parts of the network. The third advantage of TEDD is the possibility of disseminating an information only to a target area. As an example, in Figure 4, we illustrate a situation in which it is necessary to disseminate an information only to the upper right corner of the sensor field. In this case, a delivery curve is used to transport the information from the sink node to the desired area.

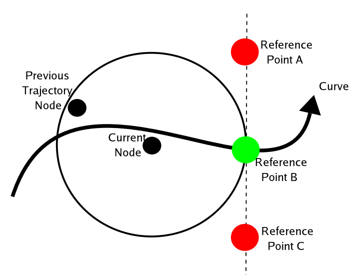



**Figura 5: Reference points.**

In TEDD, we can use two different forwarding modes. In the first one, the data is disseminated using only one flow, in a way that only one node decides to forward the packet. To minimize the number of transmissions, the node further on the curve should relay the packet. In order to determine the neighbor node that satisfies this requirement, we use reference points. As illustrated in Figure 5, when one flow is used, the node closer to reference point B relays the packet.

In the second forwarding mode, two flows are used to disseminate the data, in a way that two nodes end up forwarding the packet. As depicted in Figure 5, when two flows are used, nodes closer to reference points A and C relay the packet. It is important to point out that the choice between the two forwarding modes depends on the goal of the dissemination. There is a trade-off between maximizing the covering and minimize the number of transmissions. In situations in which minimize the number of transmissions is the most important requirement, only one flow should be used. On the other hand, when maximize the covering is the main goal, two flows should be used. As an example, only one flow should be used in the delivery curve that is used to transport an information from sink node to the target area.

In both forwarding modes, each node has to decide locally if it the one that has to forward the packet. This decision is done using a delay that is proportional to its distance to the reference point, in a way that, the node closer to the reference point is the first one to relay the packet.

Figure 6 illustrates the basic operation of TEDD. When a node receives a packet, it verifies whether it is inside the received network sector (point *A*). If it is not, it drops the packet (point *B*). If it is inside the network sector, the node verifies if its distance to the reference point is higher than the communication range (point *C*). If it is, it drops the packet. If it is not, the node waits a *delay time* (points *D* and *E*). After the node waits the *delay time*, it evaluates if any of its neighbors retransmitted the packet (point *F*). If this is the case, the node drops the packet (point *B*). Otherwise, the node calculates the reference point (to be used by the next route node) (point *G*) and, then, forwards the packet (point *H*). The goal of this scheme is to reduce the number of transmitted packets so that only the nodes closer to the reference point relay packets. These nodes are exactly those that reach the highest number of yet unreached nodes. Using this scheme, we minimize the number of transmissions, increasing the number of reached nodes.

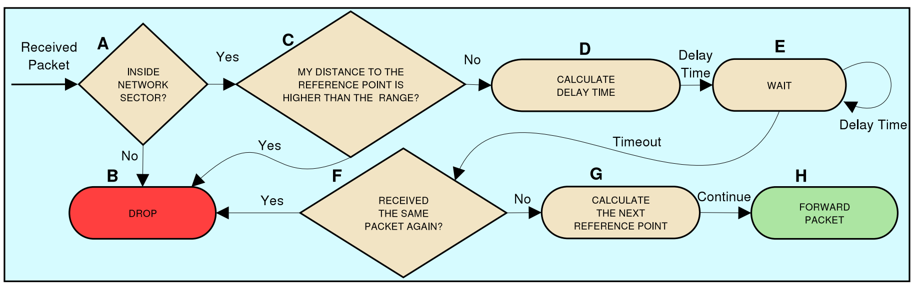



**Figura 6: Basic Operation of TEDD.**

## 5. Simulation Results

In this section, we show the TEDD behavior in two different scenarios of data dissemination. In the first one, the target area is the entire network, and in the other one, the dissemination target area is restricted to the network right upper corner. Moreover, in the last scenario, the network starts with a central low-energy area, whereas data dissemination flow should be avoided. It is assumed that each node knows its own location and the sink node knows the coordinates of all nodes in the network. One sink node is placed at the bottom left corner of the network. It performs a series of data disseminations. It is assumed that the sink node has an unlimited amount of energy. In each data dissemination, the sink node re-calculates a new set of trajectories, based on the current energy map that is obtained using the prediction-based approach, proposed in [Mini et al., 2005]. Since the cost of constructing the energy map is expected to be distributed among many applications, the cost of obtaining the map is not considered in the results. The cost of generating the trajectories is also considered irrelevant from the network perspective, since all the calculations are made at the sink node.

### 5.1. Scenario

In this section, we present the scenarios used throughout simulations. We consider a dynamic topology, where nodes periodically go into sleeping modes to save energy. The energy dissipation model used is the SEDM (State-based Energy Dissipation Model), described in [Mini et al., 2005]. Also, we consider a sensor network with static and homogeneous nodes, and battery replacement is considered to be unfeasible. Nodes are deployed randomly, forming a high-density flat topology. In order to analyze the performance of the information dissemination schemes, we implemented all protocols in the ns-2 simulator [ns2, 2002].

There are 500 nodes randomly distributed in a $35 \times 35 \text{m}^2$ sensor field. Each node has an average of 27 neighbors, being this number reduced during simulations, since some of the neighbors can be in sleeping modes. The initial energy of each node is set to $40J$ and the radio range is considered to be $5m$. Results of all simulations were obtained as an average of 33 runs of a 1000-second simulations. The sink node sends 200 broadcast messages, uniformly distributed over each simulation, to perform data dissemination through the network. The values of power consumption for each state were calculated based on Mica2 consumption [Mica2, 2004].

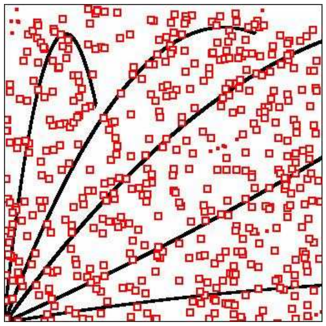
### 5.2. Dissemination to the Entire Network

In this section, a scenario where the sink node disseminates information to the entire network is studied. The behavior of TEDD is analyzed using both forwarding modes: one and two flows, presented in Section 4. Its performance is compared to the TBF (one beacon per two seconds) and to the flooding-based dissemination schemes. Both TBF and TEDD use the same trajectory generation procedure. The *maximum coverage* criterion was used to select the best set of curves. In Figures 7-a through 7-i, we show the network energy map evolution during the network lifetime. Since the *maximum number of network sectors* was set to five, this was the number of network sectors selected to maximize the network coverage[2]. Despite the fact that trajectory generation is being made dynamically, there are no significant changes in the trajectories throughout the network lifetime. This is because data dissemination occurs from the sink node to the sensor nodes and no events that cause non-uniform energy consumption in different parts of the network occur in this scenario. Together with the energy available at each node, the network coverage is shown. White squares represent nodes that receive the disseminated packets and the black ones indicate nodes that do not receive any packet at that particular moment.

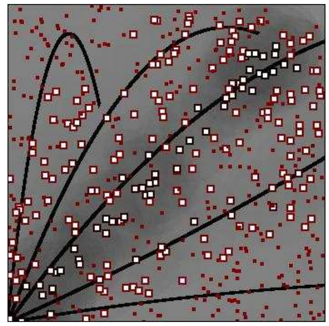
When flooding-based dissemination scheme is compared to TEDD, it can be seen that its energy consumption is significantly higher (Figures 7-d through 7-i). Although flooding starts with a better network coverage, after approximately 750 seconds of simulation, the average energy of the nodes becomes insufficient to guarantee network connectivity. As a result, network coverage drops to zero and no more packets are transmitted. This behavior is illustrated in more detail in Figures 8-a through 8-d, which show the number of reached nodes, the number of transmitted packets, the average energy of the nodes, and the number of dead nodes. The number of transmitted packets by flooding remains constant after 800 seconds, since no packets can be transmitted in a disconnected network.

Comparing the energy consumption of TBF and TEDD (one and two flows) (Figures 7-a through 7-f and Figure 8-c), the cost of neighbor table maintenance becomes evident. In average, TEDD consumes 30% less energy than the TBF. In this scenario, after approximately 850 seconds, if TBF is used, nodes located near the sink begin to die. After this threshold, TBF is not able to perform broadcasts anymore, since the sink becomes disconnected from the network. When TEDD is used, however, 90% of nodes remain alive, with more than 20% of their initial energy (Figures 8-c and 8-d) and more than 25% (two flows mode) and 20% (one flow mode) of network coverage, even after 1000 seconds of simulation (Figure 8-a). The number of transmitted packets by the

[2] In this work, the term *network coverage* is used to designate the number of nodes that receive the disseminated data.

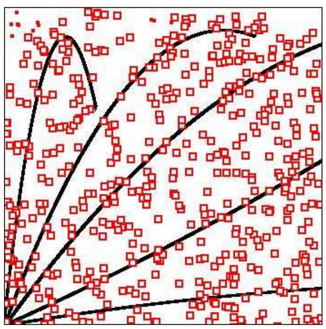


(a) TBF, time = 0s: coverage = 81%, energy = 100%.

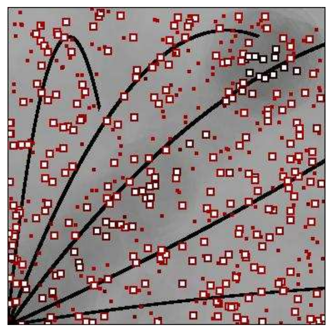


(b) TBF, time = 500s: coverage = 35.2%, energy = 45.15%.

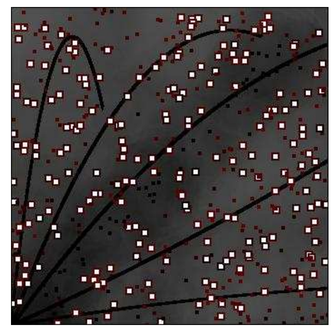
(c) TBF, time = 1000s: coverage = 0%, energy = 1.3%.



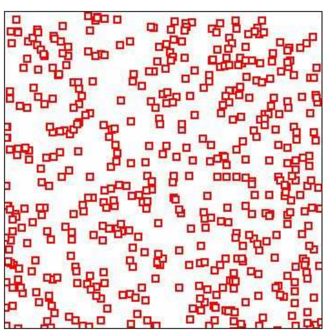
(d) TEDD, time = 0s: coverage = 97.6%, energy = 100%.



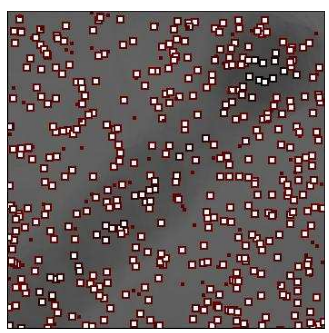
(e) TEDD, time = 500s: coverage = 52%, energy = 58.7%.



(f) TEDD, time = 1000s: coverage = 30.6%, energy = 22.5%.



(g) Flooding, time = 0s: coverage = 100%, energy = 100%.



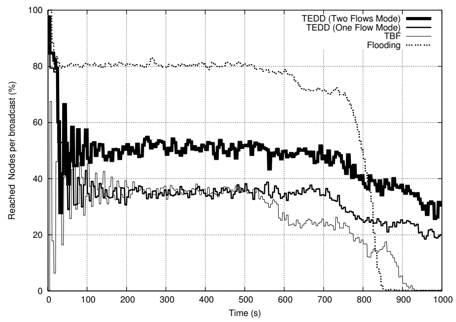
(h) Flooding, time = 500s: coverage = 80,52%, energy = 35.97%.

(i) Flooding, time = 1000s: coverage = 0%, energy = 0%.

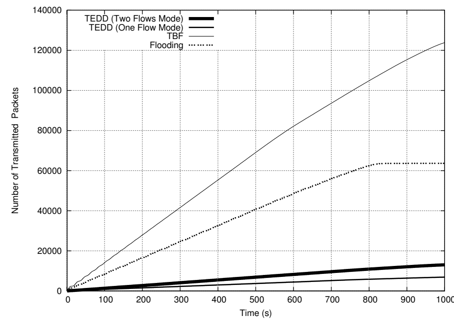
**Figura 7: Energy map and network coverage evolution (TBF, TEDD and flooding).**

TBF does not remain constant after 800 seconds in Figure 8-b, since beacon packets continue to be transmitted even in a disconnected network).

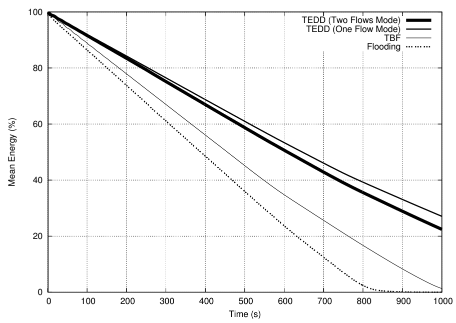
When one flow and two flow modes of TEDD are compared, it can be seen that, when two flows mode is used, TEDD achieves a 20% greater network coverage (Figure 8-a). This behavior is explained by the fact that each packet is forwarded in two different directions by each node when two flow mode is used. When one flow mode is used, however, TEDD sends less packets and, consequently, spends less energy (Figures 8-b and 8-c).



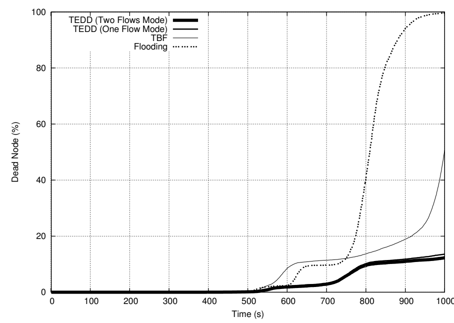

(a) Percentage of reached nodes.




(b) Number of transmitted packets.




(c) Mean energy.




(d) Percentage of dead nodes.

**Figura 8: Performance parameters (TEDD, TBF, and flooding).**

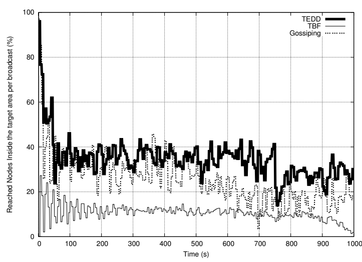
When compared to a flooding-based dissemination approach, it can be seen that, despite providing a better network coverage at first, flooding-based scheme imposes extremely high costs in terms of energy consumption. This fact compromises, firstly, the low-energy nodes and, eventually, the network lifetime as a whole. When compared to the TBF forwarding technique, it is important to point out that TEDD is a protocol that does not use neighbor tables and, therefore, spends much less energy and presents a more adaptive behavior in a dynamic topology scenario.

## 5.3. Dissemination to the Target Area

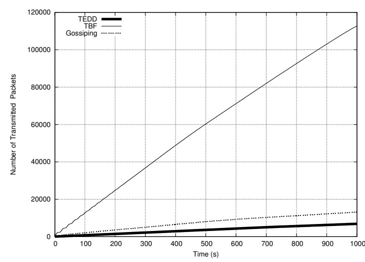
In this section, a scenario that contains an initial low-energy area, where the sink node disseminates information to a target area, is analyzed. The low-energy area is located in the middle of the network, and the target area is located at its upper right corner, as illustrated in Figure 4. The performance of TEDD is compared to the TBF and to a flooding-based dissemination scheme with probability (gossiping). In this section, the gossiping works as follows. If a node is outside the target area, the node relays packets with $0.4$ probability, otherwise, when a node is outside, it always relays the packets. In the last case, the probability is one and the gossiping is equal to flooding. Both TBF and TEDD use the same trajectory generation procedure. Outside the target area, a delivery curve connecting the sink node to the dissemination target area is generated. The one flow mode is used by TEDD to forward packets along the delivery curve. Inside the target area, the *maximum coverage* criterion was used to generate the dissemination curves, and two flows mode is used to forward packets.

Figure 9-a illustrates the network coverage inside the target area. We observe that TEDD reaches more nodes, followed by gossiping. TEDD reaches approximately $1.1$ times more nodes

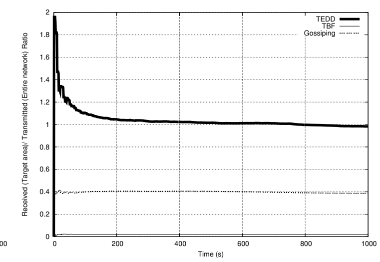
than gossiping and 2.5 times more nodes than TBF. In Figure 9-b, the number of transmitted packets in the entire network is shown. In this case, due to the cost of neighbor table maintenance, the TBF sends more packets than the other two approaches. Moreover, gossiping sends twice as much packets as TEDD. Figure 9-c shows the ratio between the number of nodes covered inside the target area and the number of packets transmitted in the entire network area. Even though the ratio achieved by TEDD is approximately one, it is significantly above the ratios achieved by the other two approaches. This apparently poor result is due to the fact that a long path has to be traveled before the packets reach the dissemination target area. In Figure 9-d, we verify that the nodes located inside the low-energy area are preserved when TEDD is used. Gossiping sends 11 times more packets than TEDD, and the TBF sends 80 times more packets than TEDD.



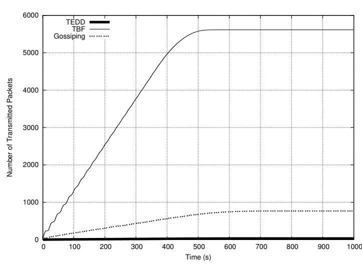

(a) Percentage of reached nodes inside the target area.



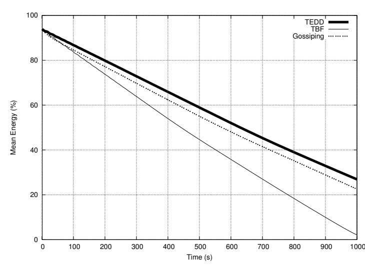

(b) Number of packets transmitted in the entire network.



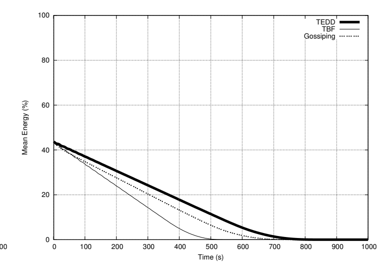

(c) Received (target area)/Transmitted (entire network) ratio.




(d) Number of packets transmitted inside the low-energy area.




(e) Mean Energy in the entire network.




(f) Mean Energy inside the low-energy area.

**Figura 9: Performance parameters (TEDD, TBF, and gossiping).**

A comparison between the energy consumption by the protocols in the entire network and in the low-energy area is shown in Figures 9-e and 9-f, respectively. In both cases, TEDD presents the least energy consumption and TBF, the greatest. The first result occurs because TEDD has a better selection mechanism of the nodes that relay data packets, and the second result is a consequence of the TBF neighbor table maintenance cost. It is important to point out that in Figure 9-f, TEDD was able to extend the nodes lifetime inside the low-energy area.

## 6. Conclusions

In this paper, we propose TEDD, a new data dissemination scheme for WSNs. The key idea is to combine concepts presented in trajectory based forwarding with the information provided by the energy map of the network. We proposed a method not only for representing the trajectories, but also for specifying them dynamically based on the energy map, which changes along the network lifetime. In TBF, nodes use a forwarding technique based on neighbor tables. This technique requires a high energy overhead to be updated, and its mechanism is not suitable for operating in dynamic topology models, where nodes can often go into sleeping modes. TEDD replaces this mechanism with a new forwarding technique: when a node receives a packet, it decides by itself if it should forward the packet based solely on its own location and the curve embedded in the packet. Simulations showed that if TEDD is used, the routing process becomes more adaptive to changes in network topology. Moreover, the energy spent with the routing activity can be concentrated on those nodes that have high energy reserves, whereas low energy nodes can be left to use their energy only to perform the sensing activity or to receive information addressed to them.

There are several improvements that we are planning to introduce to the curve generation procedure. One aspect to be explored is the way of interpreting the network. Currently, we are representing the network as a set of sensors, whose coordinates are used as input to the curve fitting procedure. Another interesting manner of performing the mapping is by viewing the network as a set of geographic points, whose energy reserves are calculated as an interpolation of the energy of those sensor nodes that cover each point. In this work, we established two criteria to select the best set of curves from all sets of curves generated using different curve types and different numbers of sectors. We plan to propose new selection criteria, possibly combining or alternating them in a dynamic fashion, depending on the application requirements.

## Referências


Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422.

Berman, M. (1983). Estimating the parameters of a circle when angular differences are known. *Journal Royal Stat. Soc.*, 32.

Estrin, D., Govindan, R., Heidemann, J., and Kumar, S. (1999). Next century challenges: scalable coordination in sensor networks. In *MOBICOM 99*, pages 263–270, USA.

Ganesan, D., Govindan, R., Shenker, S., and Estrin, D. (2001). Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM Mobile Computing and Communications Review*, 5(4).

L., L. C. and J., H. R. (1974). *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, N. J.

M., B. and Somlo, P. (1986). Efficient procedures for fitting circles and ellipses with application to sliding termination measurements. *IEEE Inst. & Meast.*, IM-35.

Mica2 (2004). Mts/mda sensor and data acquisition boards user's manual. www.xbow.com.

Mini, R. A. F., do Val Machado, M., Loureiro, A. A. F., and Nath, B. (2005). Prediction-based energy map for wireless sensor networks. *Ad Hoc Networks Journal*, 3(2):235–253.

Montgomery, D. C., Peck, E. A., and Vining, G. G. (2001). *Introduction to Linear Regression Analysis*. John Wiley & Sons.

Niculescu, D. and Nath, B. (2003). Trajectory-Based Forwarding and its Applications. In *MOBICOM 03*, pages 260–272, USA.

ns2 (2002). The network simulator. www.isi.edu/nsnam/ns.

Paige, C. C. and Saunders, M. A. (1982). Lsqr: Sparse linear equations and least squares problems. *ACM Transactions on Mathematical Software*, 8:195 – 209.

R., H. M. and E., S. (1952). Methods of conjugade gradients for solving linear systems. *Journal Res. N. B. S.*, 49:409 – 436.