

Data Dissemination in Autonomic Wireless Sensor Networks

Max do Val Machado, Olga Goussevskaia, Raquel A. F. Mini, Cristiano G. Rezende, Antonio A. F. Loureiro, Geraldo Robson Mateus, and José Marcos S. Nogueira, *Member, IEEE*

Abstract—In this paper, a new data dissemination algorithm for wireless sensor networks is presented. The key idea of the proposed solution is to combine concepts presented in trajectory-based forwarding with the information provided by the energy map of the network to determine routes in a dynamic fashion, according to the energy level of the sensor nodes. This is an important feature of an autonomic system, which must have the capacity of adapting its behavior according to its available resources.

Simulation results revealed that the energy spent with the data dissemination activity can be concentrated on nodes with high-energy reserves, whereas low-energy nodes can use their energy only to perform sensing activity or to receive information addressed to them. In this manner, partitions of the network due to nodes that ran out of energy can be significantly delayed and the network life-time extended.

Index Terms—Data dissemination, energy map, trajectory-based forwarding (TBF), wireless sensor networks (WSNs).

I. INTRODUCTION

ONE OF THE MOST important challenges in the design of wireless sensor networks (WSNs) is to deal with the dynamics of such networks. The physical world where the sensors are embedded is dynamic. Over time, the operating conditions and the associate tasks to be performed by the sensors can change. Some of the causes that may trigger these changes are the events occurring in the network, amount of resources available at nodes, particularly, energy and reconfiguration of nodes. Furthermore, it is important that sensors adapt themselves to the environment since manual configuration may be unfeasible or even impossible. In summary, the kind of distributed system we are dealing with calls for new data communication, coordination, and control algorithms for large scale, highly dynamic, and unattended WSN.

“Autonomic computing is an approach to self-managed computing systems with a minimum of human interference” [6]. Given this definition, the challenge is to design WSNs

that are self-managing, self-diagnostic, and transparent to the monitoring entity. This new computing paradigm, when applied to a WSN, means that the design of such a network should aim to embed autonomic capabilities in sensor nodes.

In WSNs, data communication, from the point of view of the communicating entities, can be divided into three cases, as depicted in Fig. 1: from sensors to a monitoring node, among neighboring sensors, and from a monitoring node to sensors. The first is used to send the sensed data to a monitoring application. The second often happens when some kind of cooperation among nodes is needed. The last, called data dissemination, is normally used to disseminate a piece of information that is important to sensor nodes. Reliable data dissemination is crucial to WSN since a monitoring node has to perform some specific 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 sensors, and send queries to the network.

In this work, an energy-efficient data dissemination protocol for WSNs, called *trajectory and energy-based data dissemination* (TEDD), is proposed. The key idea is to combine concepts presented in *trajectory-based forwarding* (TBF)¹ [15] with the information provided by the energy map² [12] to determine routes in a dynamic fashion. TEDD is comprised of two main parts. The first one is an algorithm for generating trajectories that pass through regions with higher energy reserves and avoid low-energy areas. The main idea is to select a set of nodes that are most suitable for disseminating information and to find the best set of curves passing through or near these selected points. The second part of TEDD is a new packet forwarding mechanism which is a receiver-based approach. This characteristic introduces two improvements to the TBF process. First, it eliminates the need of neighbor table maintenance, which is very expensive in terms of radio transmissions. Second, it presents a more robust behavior in dynamic topology scenarios, such as WSNs.

The rest of this paper is organized as follows. Section II discusses the related work. The two parts of TEDD are described in Sections III and IV, respectively. In Section V, we analyze the experimental results. Finally, in Section VI, we present the conclusion and the future directions.

¹Data dissemination technique in which packets are disseminated from a monitoring node to a set of nodes along a predefined curve. The main idea is to embed a trajectory in the packet, and then let the intermediate nodes forward it in a unicast manner to those nodes that lie close to the trajectory.

²Energy map is the information about the amount of energy available at each part of the network.

Manuscript received October 18, 2004; revised May 5, 2005. This work was supported in part by CNPq, Brazil, under Process Number 55.2111/2002-3, SensorNet Project (<http://www.sensornet.dcc.ufmg.br>), and in part by Scholarships from CAPES, Ministry of Education, Brazil.

M. do Val Machado, O. Goussevskaia, C. G. Rezende, A. A. F. Loureiro, G. R. Mateus, and J. M. S. Nogueira are with the Department of Computer Science, Federal University of Minas Gerais, Belo Horizonte 30123-970, Brazil (e-mail: maxm@dcc.ufmg.br; olga@dcc.ufmg.br; rezende@dcc.ufmg.br; loureiro@dcc.ufmg.br; mateus@dcc.ufmg.br; jmarcos@dcc.ufmg.br).

R. A. F. Mini is with the Department of Computer Science, Pontifical Catholic University of Minas Gerais, Belo Horizonte 30535-610, Brazil (e-mail: raquel@dcc.ufmg.br).

Digital Object Identifier 10.1109/JSAC.2005.857209

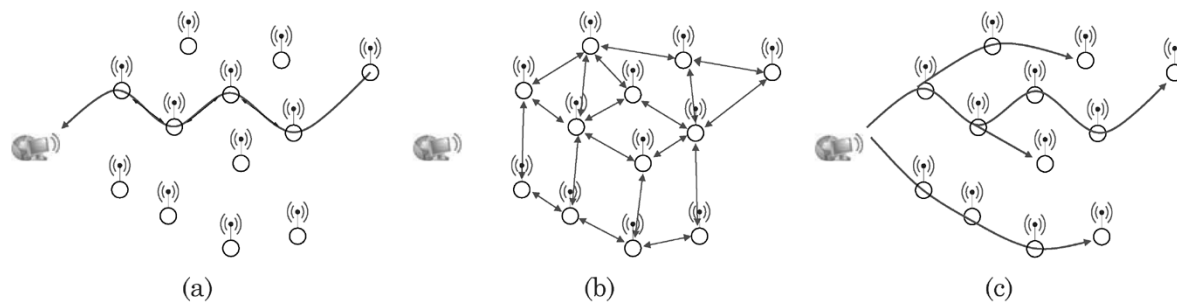


Fig. 1. Data communication schemes in WSNs. (a) Data communication from a sensor node to a monitoring node. (b) Data communication among neighboring nodes. (c) Data communication from a monitoring node to sensor nodes.

II. RELATED WORK

Several different routing protocols for WSNs have been proposed in the literature [1], [3], [7], [9]. Among all algorithms already proposed in the literature, the closest to the one presented in this work is the TBF [15], that is a technique to disseminate messages in dense wireless networks. The key idea is to embed a curve (trajectory) in the packet to be disseminated from a monitoring node to sensor nodes [Fig. 1(c)], and then let the intermediate nodes forward it in a unicast manner to those nodes that lie close to the curve. TBF is a sender-based algorithm since the current node systematically chooses the next hop of the route. This forwarding decision is based on the curve and a neighboring table. To update this table, nodes exchange beacon packets periodically. TBF is a source routing protocol since the entire trajectory is defined by the data dissemination source. In traditional source routing protocols for mobile ad hoc networks [8], the source node inserts all nodes of the path into the packet as a discrete set of points, generating a considerable overhead and making impracticable its use in WSNs. Two main advantages of TBF are compact representation of a route, since curves can be described using few parameters, and node independence, since no particular node address is specified in the trajectory.

Algorithm 1 shows the basic operation of TBF.³ When a node receives a beacon packet, it updates its neighbor table. If the received packet is not a beacon, but a data packet, this node checks if it is the node elected to forward this packet. If it is not the case, the node drops the packet, otherwise it chooses the next node in the trajectory defined by the curve. This choice is made based on its neighbor table and a predefined forwarding policy (e.g., minimum deviation). After choosing the next node, the current node transmits the packet.

Algorithm 1: TBF-Receiving packet

```

input: the received packet
if the packet is a beacon then
  Update my neighbor table
else
  /* The received packet is a data
  packet*/
  if I am the node elected to forward
  this received data packet then
    Choose the next node in the
    trajectory

```

³Since TBF is a source routing protocol, its basic operation is similar to the traditional source routing protocols. The main difference to them is that TBF defines the routes as curves.

```

  Insert the chosen node as the next
  hop

```

```

  Transmit the packet

```

```

else

```

```

  Drop the packet

```

```

endif

```

```

endif

```

Despite its advantages, TBF has two main drawbacks. First, the overhead required to update the neighbor tables increases the number of transmitted packets, and consequently, the total energy spent. In dynamic topology environments, such as WSNs in which nodes frequently enter a sleep mode to save energy, mechanisms for neighbor table maintenance have a prohibitive cost. Second, TBF is not fault tolerant in scenarios in which topological changes are faster than the neighbor table updates. In this case, broken trajectories happen when the selected node is unavailable (e.g., the node is sleeping). Therefore, we note a tradeoff between the neighbor table update overhead and the protocol robustness.

III. DYNAMIC TRAJECTORY GENERATION

In this section, we discuss the problem of generating trajectories for data dissemination. First, the problem is defined and, afterward, the proposed solution is presented.

A. Problem Definition

As input to this problem, we have a data forwarding protocol, a set of nodes distributed in an ad hoc manner over the WSN, and a monitoring node that disseminates data. The problem of generating routing trajectories asks for the ideal number n of trajectories and the $n \times p$ parameters (we suppose a continuous model, describing a curve using p parameters), such that the objectives of the routing protocol are achieved or maximized.

Despite providing several insights into the problems that might arise during the process of specifying a forwarding trajectory, the authors of TBF [15] do not present solutions to the curve generation problem, specially to the problem of how and based on what kind of information the trajectories should be generated for the routing. In Cartesian routing [5], the route is defined as a straight line between the router and the destination. In source-based routing [8], the route is specified as a discrete set of points. In other protocols [3], [7], the route is “discovered” instead of being defined. Therefore, to the best of our knowledge, the solution proposed in this work is pioneer.

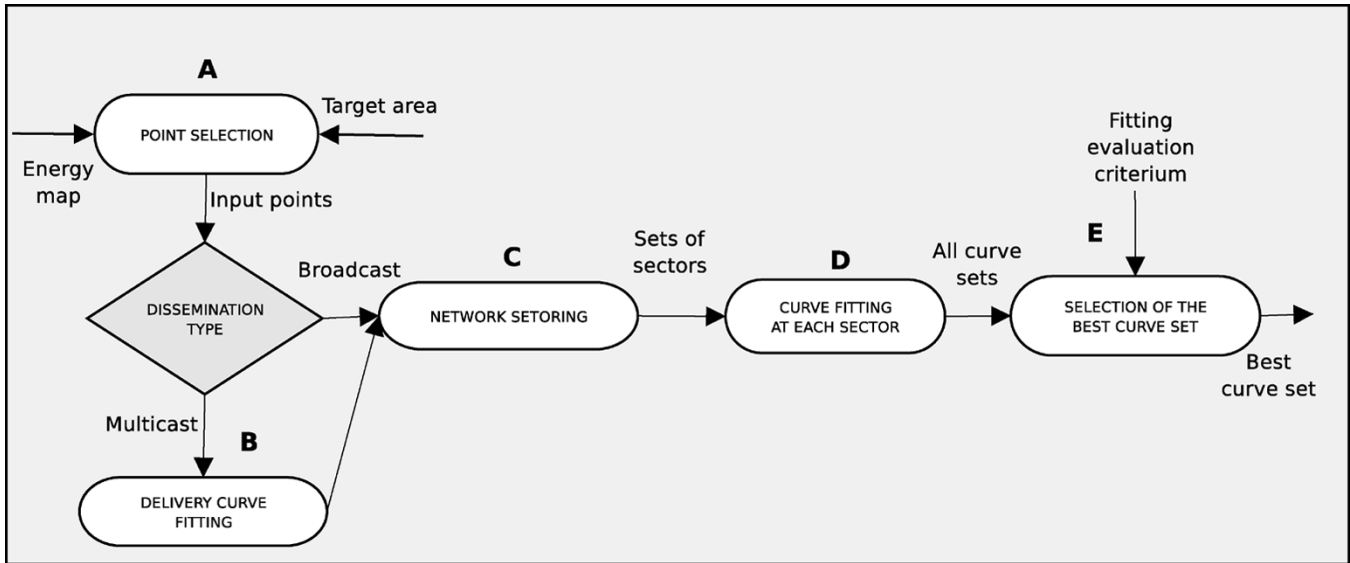


Fig. 2. Process of trajectory generation.

The problem of generating trajectories can be divided into several subproblems. In the following, we discuss and propose solutions to each subproblem.

B. Input: The Energy Map

The first question that arises when solving the problem of curve generation is: what kind of information should the procedure be based on. In this work, we use the energy map as our input, since energy is an important constraint in WSNs. In [12], the authors analyze the cost of obtaining this map using a prediction-based approach and show that it is viable in WSNs. It is worth mentioning that the cost of obtaining the energy map can be amortized among different network applications, and, thus, neither of them has to pay for this information itself.

C. Point/Node Selection

Having defined the input to the procedure (i.e., the energy map), a subset of points⁴ from the WSN has to be selected to serve as input data for the fitting process. Several strategies can be used to select this subset. 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 nodes with little energy to participate in the forwarding process. Another criterion is the node density in each part of the network. The denser the region the trajectory passes, the greater the network connectivity is, and the better the chances of the packet to be delivered successfully. This occurs because nodes are frequently programmed to turn off their radios in order to save energy. Therefore, there is always a possibility of the trajectory to break, in case there are no nodes “awake” to propagate the packet.

In this work, the points are selected using a combination of energy and density criteria. For every node, the sum of energies of all its neighbors, together with its own is calculated. Afterwards, the nodes are sorted in decreasing order of this factor,

⁴We can study a WSN as a set of points where each point can be a node in the network graph or a pixel in the image of the energy map.

and the first half of the nodes is selected to be the input to the fitting procedure.⁵ In this manner, the trajectories are “forced” to pass through regions of higher densities and energy reserves. The main concern here is to avoid that low-energy nodes participate in forwarding activities and to minimize packet losses due to broken trajectories.

D. Curve Representation and Curve Fitting

The second question that arises when solving the problem of trajectory generation is: which model should be used to represent the trajectories. In this work, we use a polynomial representation that offers a compact encoding, allowing to control the number of parameters by limiting the degree of the polynomial. Moreover, the value of the dependent variable y is directly computed for any value of the independent variable x . Based on experiments, we could observe that this type of representation is flexible and expressive enough to generate trajectories that avoid low-energy and low-density areas.

Having defined the model to represent the trajectories, a curve-fitting algorithm has to be determined. Due mainly to its simplicity, we use multiple linear regression [10], [14] to fit the curves into a set of points and, in particular, we use the LSQR algorithm⁶ [17].

E. Architecture

Having discussed the previous subproblems, some questions related to the architecture of the curve generation process have to be answered. The architecture proposed in this work is illustrated in Fig. 2 and is described in the following.

The process has some variations depending on the dissemination type. As showed in Fig. 2 (Point A), the first step is to select points from the dissemination target area. If the dissemination is

⁵The reason to select only 50% of nodes is purely empirical. Experiments were made with different percentages, and better results were obtained by eliminating 50% of nodes from the fitting procedure.

⁶The computational requirements of LSQR are: storage $(n + 2p)$, and number of floating-point multiplications per iteration $(3n + 5p)$. The maximum number of iterations was set to $4np$.

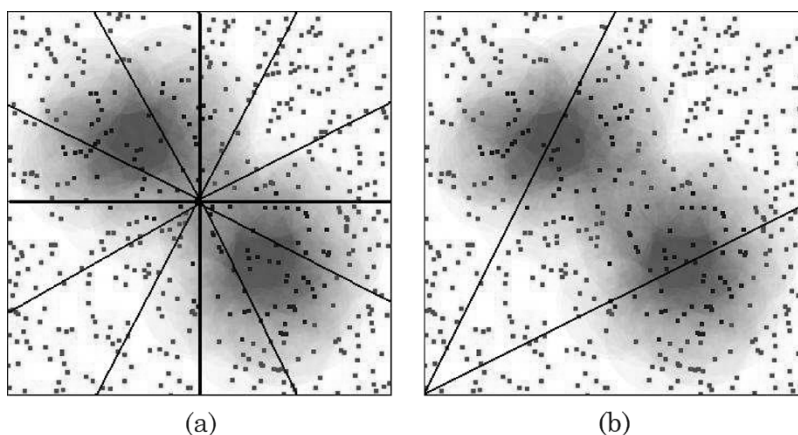


Fig. 3. Maximum number of sectors depends on the position of the monitoring node. (a) Network sectoring with monitoring node in the center. (b) Network sectoring with monitoring node at the corner.

a broadcast, the points are selected from the whole map, which is the target area. If the dissemination is a multicast (the target area is a subset of the map), two sets of points are selected: one inside the area between the monitoring node and the target area, and the other inside the target area. The former set is used to fit a special curve, called *delivery curve* [Fig. 2 (Point B)], that is used as a tunnel between the monitoring node and the target area. The latter set is used to fit curves inside the target area. The delivery curve must intersect the monitoring node at one end, and the target area at another. To not overload the nodes located at the point of intersection between the delivery curve and the target area, this point must be defined dynamically, based on energy/density. It means that, given all points located on the target area boundary, the intersection point is the one with greater energy/density factor. Inside the target area, the origin of the generated curves is the intersection point. A procedure called *network sectoring* is used to define the ideal number of curves inside the target area [Fig. 2 (Point C)], as explained in this next section.

1) *Network Sectors*: Given a set of points that we would like to force to participate in the forwarding process and given the curve type (polynomial), 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 monitoring node, the problem of determining the best number of curves 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 [Fig. 2 (Point D)].

An arbitrary number of network sectors could be used. However, it is not reasonable to have a large value, since this would result in an unacceptably high number of parameters to be transmitted with each packet and an unacceptably low number of points at each sector, compromising the quality of the fitting procedure. A maximum limit can, therefore, be defined for the number of network sectors. This limit depends on the position of the monitoring node. If it is located at one of the corners of the target area, the sectoring is made within a 90° angle. If it is

located at the center, the sectoring is made within a 360° angle, allowing the greatest possible number of sectors. These situations are illustrated in Fig. 3.

Besides the number of sectors, the degree of the polynomial also has influence on the quality of the fitting procedure. Therefore, the curve fitting is not only performed for different numbers of network sectors, but also for different polynomial degrees. The maximum polynomial degree used in this work is four, since the higher the degree of the polynomial, the greater the complexity of calculating the distance between each node and the curve.

Given a maximum number of network sectors and a maximum polynomial degree, all possible curve sets are generated. By selecting the curve set with the “best” *average quality*, we determine the boundaries and the angles of the sectors, as is explained in the next section.

2) *Best Curve Set Selection*: The last step in the curve generation process is the selection of the best curve set, as shown in Fig. 2 (Point E). This selection 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 one set 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 quality of one curve can be calculated based on different criteria, depending on the application requirements. In this work, the following fit evaluation criteria were used: *maximum average energy*, which maximizes the average energy of the nodes within the covering range of the curve ($distance(node, curve) \leq node_radio_range$), and *maximum coverage*, which maximizes the total number of nodes within the covering range of the curve. Finally, the “best” fit quality is determined by calculating the average criteria of each set, and selecting the one with the highest average fit quality.

In the following section, we provide examples of network sectoring and curve fitting in different network scenarios.

3) *Examples of Network Sectoring*: Fig. 4(a) and (b) illustrates two sets of broadcast curves, selected for two different energy maps, using the *maximum average energy* criterion. Fig. 4(c) and (d) illustrates the same scenarios, however, using the *maximum coverage* criterion. It can be observed that when

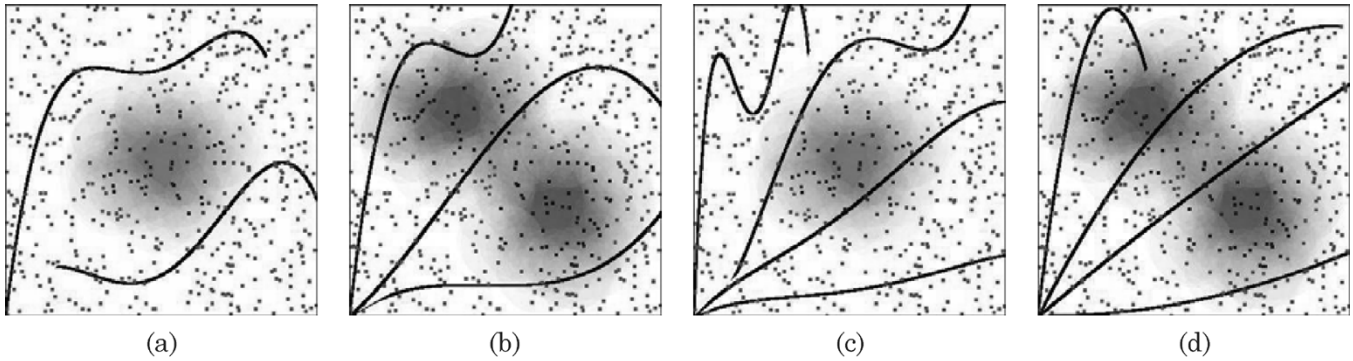


Fig. 4. Curve sets to perform broadcasts. Energy map of an area of $35 \times 35 \text{ m}^2$ with one and two low-energy areas. (a) Maximum energy. (b) Maximum energy. (c) Maximum coverage. (d) Maximum coverage.

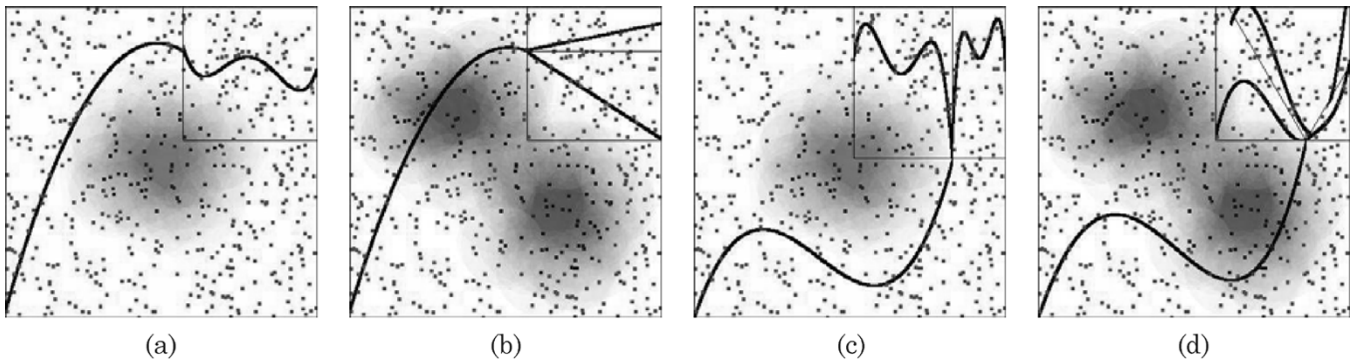


Fig. 5. Curve sets to perform multicasts. Energy map $35 \times 35 \text{ m}^2$ with one and two low-energy areas. Target area = $(20, 20) - (35, 35)$. (a) Maximum energy. (b) Maximum energy. (c) Maximum coverage. (d) Maximum coverage.

the first criterion is used, a set with fewer sectors is selected, and the curves avoid the low-energy areas. When the second criterion is applied, the maximum number of sectors is selected, and the curve inside each sector is fitted closer to the nodes with greater energy reserves.

Fig. 5(a) and (b) illustrates two sets of curves generated to perform a multicast, using the *maximum average energy* criterion. The target area is determined as a rectangle with coordinates $(20,20)-(35,35)$. Fig. 5(c) and (d) illustrates the same scenarios, however using the *maximum coverage* criterion. It can be observed that the delivery curve avoids low-energy areas. When the first criterion is used, less sectors are used inside the target area. When second criterion is applied, more sectors are used.

F. Some Remarks

It is important to point out that the trajectory generation strategy proposed here is not restricted to the illustrated network scenarios. An energy map of a network with an arbitrary shape and an arbitrary number of randomly distributed monitoring 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 monitoring nodes. This solution presents two important features of an autonomic system: flexibility and adaptability.

Another relevant consideration is about the process of encoding the trajectories. Curve parameters can be embedded in the packet header or can be preconfigured in the nodes before delivering them. However, in the latter case, the monitoring node should be able to update those values periodically.

IV. PACKET FORWARDING POLICY

In this section, we present the second part of our solution that consists of the data dissemination model of TEDD, whose goal is to discover the best energy-efficient routes.

A. Proposed Improvements

TEDD extends the principles of TBF by incorporating the usage of the energy map. The proposed protocol defines a receiver-based data dissemination policy, i.e., each node upon receiving a packet decides itself whether to relay it or not, as opposed to TBF that is a sender-based data dissemination policy. In TEDD, the decision to forward a packet or not is based on the node geographical location and the packet information. The forwarding decision process uses a temporization policy: before relaying a packet, the current node waits a small time interval. After this time, if no neighbor has relayed the packet, the node transmits it. The key idea of this technique is how to estimate the delay time, based only on the distance between the current node and a point ahead on the curve called *reference point*.

Using the temporization policy, TEDD overcomes the drawbacks of TBF. First, TEDD avoids both the necessity of neighbor tables and beacon transmissions, and consequently, spends less energy in the forwarding process. In TBF, the neighbor table is fundamental to the process of choosing the next hop of the trajectory. Second, TEDD is more robust than TBF because nodes are not selected by the previous elected node. TEDD is a receiver-based protocol and, thus, avoids situations where the forwarding process is interrupted because the selected node is un-

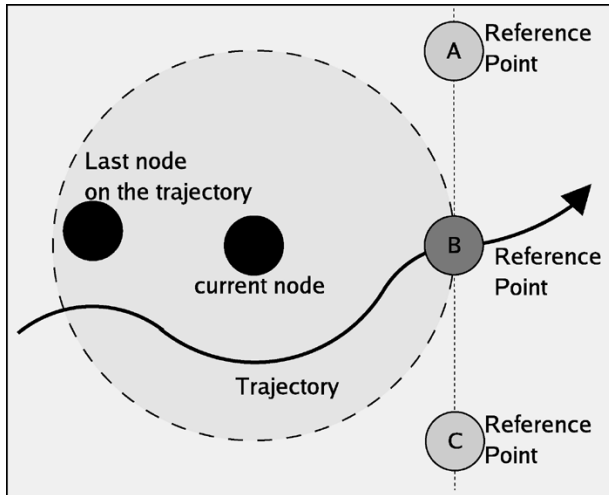


Fig. 6. Reference points.

available. Another important advantage of TEDD is the possibility of disseminating data only to a target area, as shown in Fig. 5. In this case, the protocol avoids nodes not interested in this data (outside the target area).

B. Temporization Policy and Forwarding Modes

The proposed temporization policy is based on the distance from the current node to a point ahead on the curve, called reference point. In particular, the reference point is the point (not necessarily a node) closest to the curve localized at the circumference with center at the current node and radius equals to the node communication radius (Fig. 6). In each relay, the selection of the reference point is determined by the previous hop of the trajectory. Each node that receives a packet adjusts its delay time using its distance to the reference point sent in the packet.

Based on this policy, TEDD defines two different forwarding modes: one data flow and two data flows. In the first one, the data is disseminated using only one flow, in a way that only one node decides to forward 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 illustrated in Fig. 6, when one flow is used, the node closer to reference point B relays the packet; and when two flows are used, the node closer to reference point A and the one closer to reference point C relay the packet. As an example, in Fig. 5(a)–(d), TEDD uses one flow in the delivery curve and two flows in the target area.

The choice between one or two flows depends on the goal of the dissemination. In data dissemination protocols, there is a tradeoff between minimizing the number of transmissions and maximizing the coverage. In situations in which the first goal is the most important requirement, only one data flow should be used. On the other hand, when maximizing the coverage is the main goal, two data flows should be used. Its important to point out that independently of the number of data dissemination flows, the previous trajectory node always selects only one reference point. In the two flows forwarding mode, nodes that receive a packet calculate the two reference points also using the reference point present in the packet.

C. Basic Operation

Algorithm 2 presents the basic operation of TEDD. When a node receives a packet, it verifies whether it is inside the received network sector. If it is not, it drops the packet. If it is inside the network sector, the node verifies if its distance to the reference point (calculated by the previous hop) is higher than the communication range. When the forwarding mode with two flows is used, the node calculates the two new reference points and its distance to both points, and then selects the closest point as the reference point. If the calculated distance is higher than the communication range, the node drops the packet. If it is not, the node waits a delay time that is calculated according to its distance to the reference point. The smaller the distance, the smaller the delay. After the node waits the delay time, it verifies whether any of its neighbors retransmitted the packet. If this is the case, the node drops the packet. Otherwise, the node selects the reference point and, then, forwards the packet. The process of selecting the reference point is presented in the next section.

Algorithm 2: TEDD–Receiving packet

```

input: the received packet
if I am inside the received network
sector then
  Calculate my distance to the
  reference point
  if this distance vale is less or
  equal to the communication range then
    Calculate the delay time
    Wait the delay time
    if any of my neighbors retransmitted
    this packet then
      Drop the packet
    else
      Calculate the reference point
      (Algorithm 3)
      Forward the packet
    endif
  else
    Drop the packet
  endif
else
  Drop the packet
endif

```

D. Selecting the Reference Point

The process of selecting a reference point is described in Algorithm 3 and it is determined by the previous hop in the trajectory. Before forwarding the packet, the node calculates two special points on the curve, $P'(x', y')$ and $P''(x'', y'')$, where $x' = x_{\text{node}}$ and $x'' = x_{\text{node}} \pm \text{communication_range}$. In this case, x_{node} and $\text{communication_range}$ are, respectively, the x -coordinate and the communication range of the node that is selecting the reference point. Similarly, the same holds for the values of y' and y'' . If the data dissemination is from left to right, $x'' = x_{\text{node}} + \text{communication_range}$, otherwise, if it is from right to left, $x'' = x_{\text{node}} - \text{communication_range}$. In the second step, the algorithm defines the straight line $y = ax + b$

TABLE I
DEFAULT VALUES USED IN SIMULATIONS

Parameters	Value	Parameters	Value
Number of nodes	500	Sensor field size	35×35m ²
Initial energy	40 J	Power consumption	Mica2 [11]
Communication range	5m	Simulation Time	1000 seconds
Number of Disseminations	1 each five seconds	TBF beacon ratio ^a	0.5 beacon/second

^aThis value is empirical. Experiments were made with different beacon rate, and better results were obtained using 0.5 beacon/second.

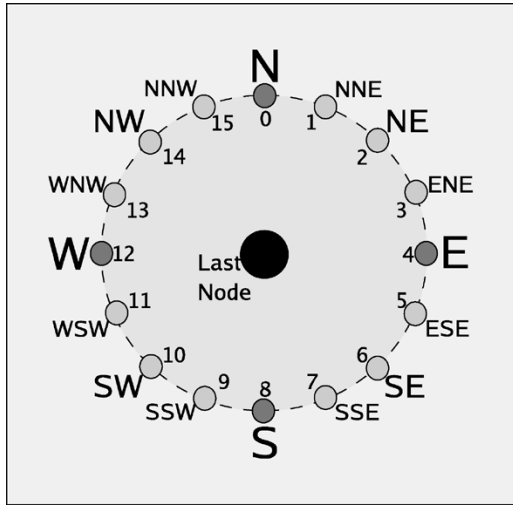


Fig. 7. Possible reference points.

that passes through P' and P'' . This equation is used by the proposed algorithm instead of the curve equation because the algorithm has to calculate the distance from a curve to a point. The evaluation of the curve/point distance is not trivial, mainly considering the limited resources of sensor nodes. On the other side, the distance from a straight line to a point is easier to be calculated. This heuristic presents good results, since the generated curves do not present a great variation inside the node communication radius. In the third step, the node determines the quadrant where the reference point is localized. The quadrant of a reference point is an important concept because it reduces the amount of possible reference points to examine. The communication radius has four quadrants, the first one is located at the northeast and the others are, respectively, located at the northwest, the southwest, and the southeast. Moreover, this concept is the same one used in the plain geometry for the Cartesian plan, except that the source is the current node. The quadrant of a reference point is obtained using Algorithm 4 and it is detailed below. In the last step, the node selects the reference point among some points located in the selected quadrant. In Fig. 7, the points of each quadrant are, respectively: (N, NNE, NE, ENE, E); (N, NNW, NW, ENW, W); (W, WSW, SW, SSW, S); and (E, ESE, SE, SSE, S).

Algorithm 3: TEDD—Selecting the reference point

input: the received packet
Calculate points P' and P'' where $x' = x_{node}$
and $x'' = x_{node} \pm communication_range$

Calculate the straight line $y = ax + b$ that passes through points P' and P''
Call the procedure to discover the quadrant of the reference point (Algorithm 4)
Select the reference point using the quadrant and the straight line $y = ax + b$.
Return the reference point

The process of discovering the quadrant of the reference point is described in Algorithm 4 and it is the third step of Algorithm 3. Using quadrants, TEDD reduces the amount of possible reference points, and consequently, the cost of selecting this point. As illustrated in Fig. 7, the use of quadrants reduces this value from sixteen to five. The proposed process considers two scenarios: 1) the curve intercepts the “communication circle” and 2) it does not intercept. In the former, the desired quadrant is the one that contains the reference point; in the latter, the desired quadrant is the one nearest to the curve. To identify whether the curve intercepts the communication radius, TEDD uses the following procedure. The node verifies if the point P' is inside its communication radius. If it is, the selected quadrant is the one where P'' is inside. In this case, TEDD knows that the circumference is intercepted by the line $y = ax + b$ and P'' is inside the quadrant of the reference point. On the other hand, if P' is outside the communication circle, TEDD evaluates the inclination a of the straight line $y = ax + b$, and the values of y' (the y -coordinate of point P'') and y_{node} (the y -coordinate of the node that is selecting the reference point). TEDD selects the first quadrant when $y_{node} \geq y'$ and $a > 0$. The other quadrants are, respectively, selected when: $y_{node} < y'$ and $a > 0$ (second); $y_{node} \geq y'$ and $a \leq 0$ (third); and $y_{node} < y'$ and $a \leq 0$ (fourth).

Algorithm 4: TEDD—Discover the quadrant of the reference point

input points P' and P'' , and straight line $y = ax + b$;
if P' is localized inside my communication circle **then**
 I select the quadrant where point y'' is localized.
else
 if $y_{no} \geq y'$ **then**
 if $a > 0$ **then**
 I select the first quadrant.
 else
 I select the third quadrant.

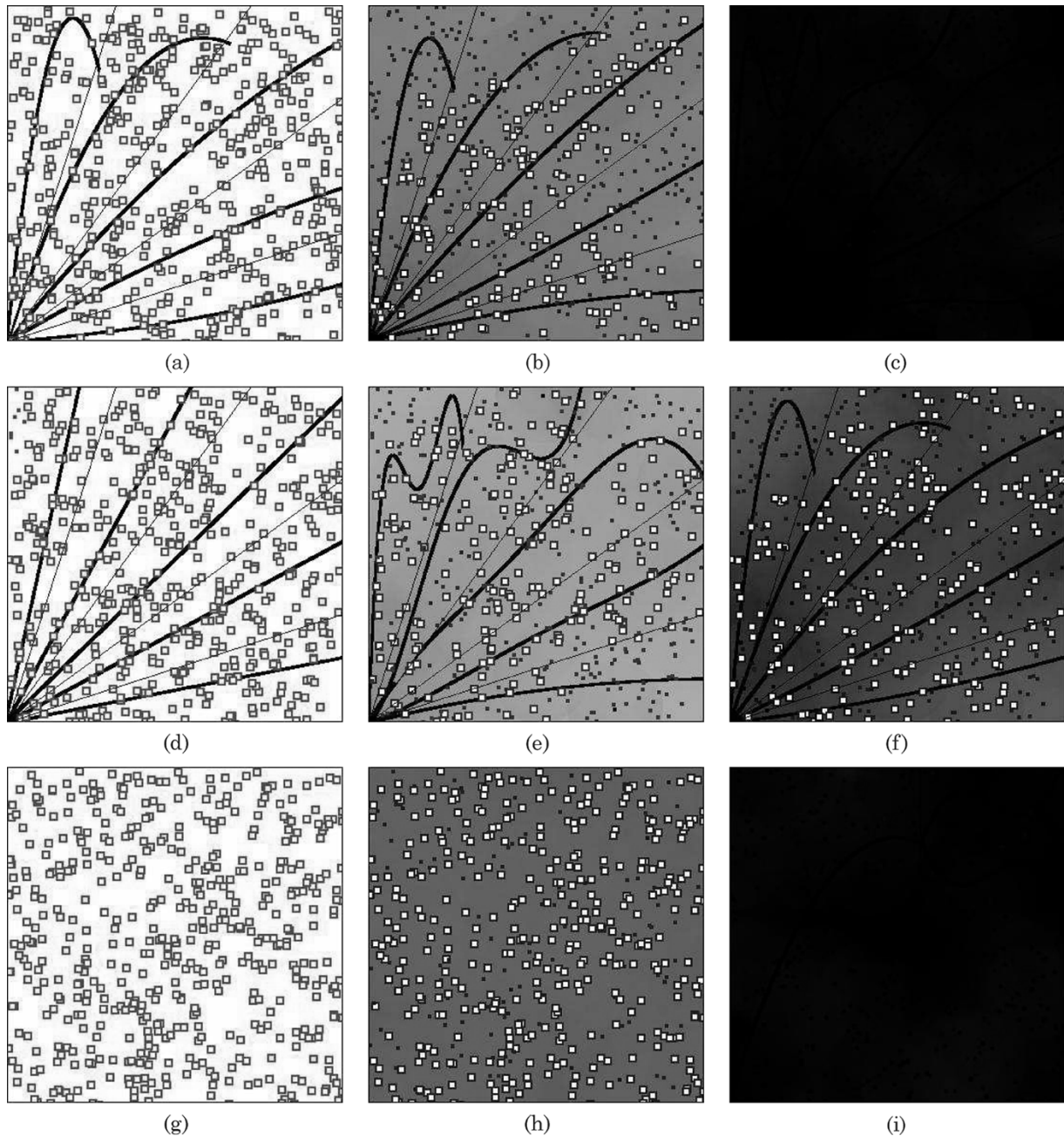


Fig. 8. Energy map and network coverage evolutions for one instance of each protocol evaluated (TBF, TEDD with one flow forwarding mode and *flooding*) in a broadcast scenario. (T = time, C = coverage, E = mean energy). (a) TBF, $T = 0$ s: $C = 100\%$, $E = 100\%$. (b) TBF, $T = 500$ s: $C = 39\%$, $E = 48\%$. (c) TBF, $T = 1000$ s: $C = 0\%$, $E = 1.3\%$. (d) TEDDc(1F), $T = 0$ s: $C = 100\%$, $E = 100\%$. (e) TEDD(1F), $T = 500$ s: $C = 51\%$, $E = 60\%$. (f) TEDD(1F), $T = 1000$ s: $C = 46\%$, $E = 23\%$. (g) Flooding, $T = 0$ s: $C = 100\%$, $E = 100\%$. (h) Flooding, $T = 500$ s: $C = 79\%$, $E = 38\%$. (i) Flooding, $T = 1000$ s: $C = 0\%$, $E = 0\%$.

```

end if
else
  if  $a > 0$  then
    I select the second quadrant.
  else
    I select the fourth quadrant.
  end if
end if
end if
end if

```

E. Some Remarks

The goal of TEDD is to reduce the number of transmitted packets so that only nodes closer to the reference point relay packets. Moreover, TEDD maintains a good network coverage since nodes closer to the reference point are exactly those that reach the highest number of yet unreached nodes. Using this algorithm, we are able to reach our goals and, thus, increase the *received/transmitted* ratio, which is an important metric to indicate the quality of a data dissemination technique. One drawback of this approach is a higher latency to deliver data. This

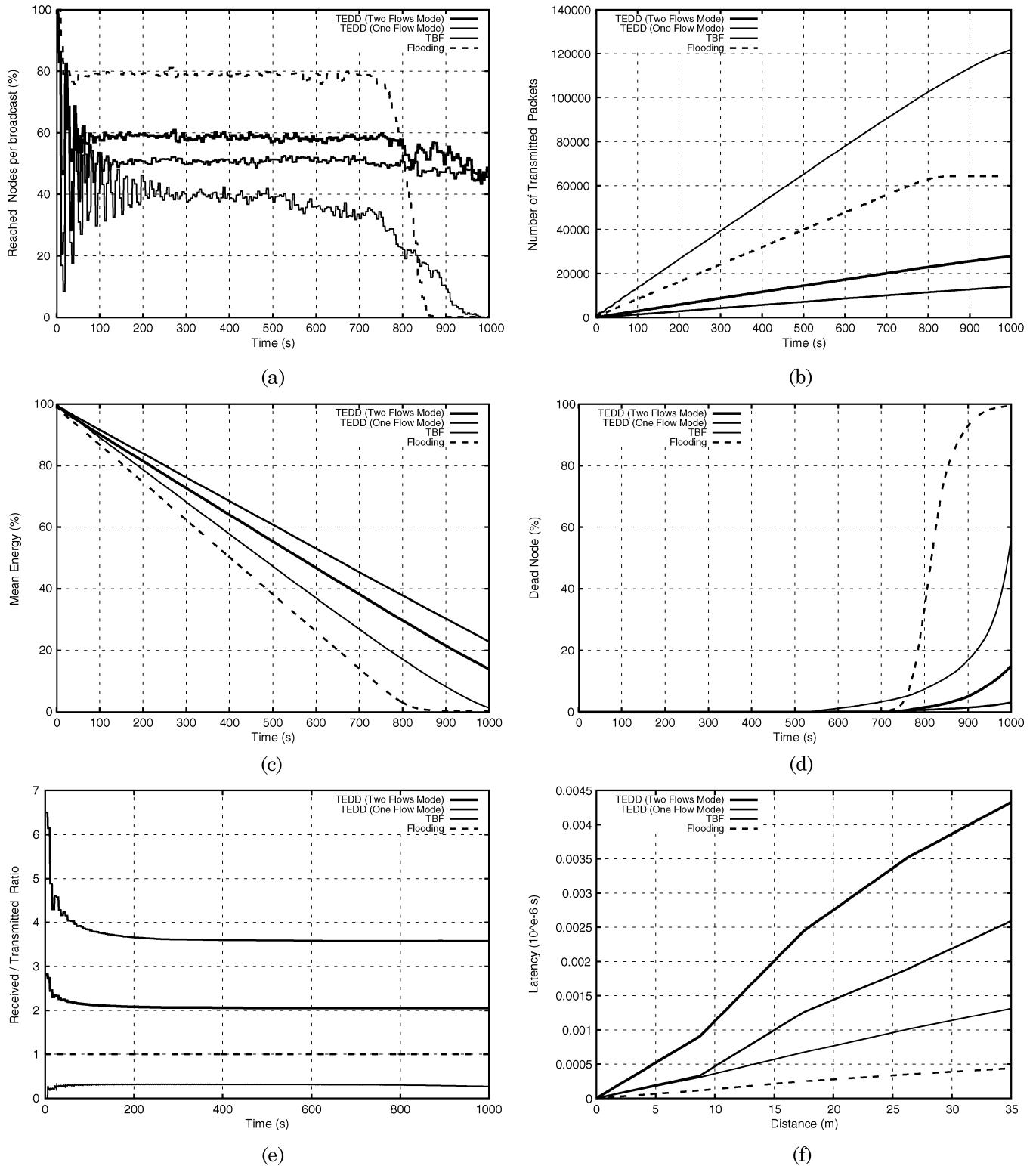


Fig. 9. Performance parameters (TEDD, TBF, and flooding) in a broadcast scenario. (a) Percentage of reached nodes. (b) Number of transmitted packets. (c) Mean energy. (d) Percentage of dead nodes. (e) *Received/transmitted* ratio. (f) Latency.

and other metrics are evaluated using simulations, as discussed in the next section.

V. SIMULATION RESULTS

In this section, we show the behavior of TEDD in two scenarios of data dissemination in a WSN. In the first one, the

monitoring node disseminates data to the entire network. In the second one, it disseminates data to a target area located at the right top corner of the sensing field, which has an initial low-energy area in the center of it. The remainder of this section is organized as follows. Section V-A presents the simulation parameters. Sections V-B and V-C show our protocol performance in both data dissemination scenarios.

TABLE II
AVERAGE NUMBER OF OPERATIONS, TRANSMISSIONS, AND NETWORK COVERAGE AT EACH DATA
DISSEMINATION FOR TEDD, TBF, AND GOSSIPING IN A MULTICAST SCENARIO

Protocols	Operations			Comparisons	Assignments	Tx	Rx
	+	-	× /				
TEDD(one flow)	10.2	13.9		21.2	4.5	70	50.2%
TEDD(two flows)	19.4	21.6		36.7	6.6	139	57.4%
TBF	3.8	3.1		46.3	26.0	573	32.4%
Flooding	-	-		42.3	0.7	321	64.2%

A. Scenarios

In this section, we present the scenarios used throughout the simulations. We consider a dynamic topology, where nodes are static but periodically go into a sleeping mode to save energy, which leads frequently to topology changes. In [4], Hill *et al.* state that a WSN should embrace the philosophy of getting the work done as quick as possible and going to sleep. The best way to save energy is to turn off parts of the sensor that are not needed, as modeled by a state-based energy dissipation model (SEDM) presented in [13]. In order to analyze the performance of TEDD, we use the ns-2 simulator [16].

We consider a WSN with static and homogeneous nodes with a finite amount of energy. Nodes are deployed randomly, forming a high-density flat topology. It is assumed that each node knows its own location and the monitoring node knows the coordinates of all nodes. One monitoring node with no energy, memory or processor restrictions is placed at the bottom left corner of the network and performs a series of data disseminations. In each data dissemination, a new set of curves is recalculated based on the current energy map that is obtained using a prediction-based approach [12]. The cost of obtaining this map is not considered in the results since it is expected to be distributed among different network activities. The cost of generating the curves is also not considered since they are generated in the monitoring node. The numerical values chosen for the simulations can be seen in Table I.

B. Dissemination to the Entire Network

In this section, a scenario where the monitoring 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 IV. Its performance is compared to the TBF and to the flooding-based dissemination schemes. Both TBF and TEDD use the same trajectory generation procedure, described in Section III. The *maximum coverage* criterion was used to select the best set of curves. In Fig. 8, we show the network energy map evolution during the network lifetime. 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 packets at that particular moment. Since the *maximum number of network sectors* was set to five, this was the number of network sectors selected to maximize the network coverage.⁷

⁷In this work, the term *network coverage* is used to designate the number of nodes that receive the disseminated data.

When we compare a flooding-based dissemination scheme to TEDD, we can see that its energy consumption is significantly higher [Fig. 8(d)–(i)]. Although flooding starts with a better network coverage, after approximately 750 s of simulation, the average node energy becomes insufficient to guarantee network connectivity. As result, network coverage drops to zero and no more packets are transmitted. This behavior is illustrated in more detail in Fig. 9(a)–(d), which show the number of reached nodes, the number of transmitted packets, the average node energy, and the number of dead nodes, respectively. The number of transmitted packets by flooding remains constant after 800 s, since no packets can be transmitted in a disconnected network. Moreover, in Fig. 9(a), we observe that flooding covers only about 80% of the network. It happens because of the dynamic topology, i.e., nodes periodically go into sleeping mode to save energy.

Comparing the energy consumption of TBF and TEDD (both forwarding modes: one and two flows) [Fig. 8(a)–(f) and Fig. 9(c)], the cost of neighbor table maintenance becomes evident. In average, TEDD consumes 22% less energy than TBF. In this scenario, after approximately 600 s, if TBF is used, nodes located near the monitoring node begin to die. After 950 s, TBF is not able to perform broadcasts anymore, since the monitoring node becomes disconnected from the network. When TEDD is used, however, more than 98% (one flow) and 82% (two flows) of nodes remain alive, with more than 21% (one flow) and 16% (two flows) of their initial energy [Fig. 9(c)–(d)] and more than 45% of network coverage, even after 1000 s of simulation [Fig. 9(a)]. The number of transmitted packets by the TBF does not remain constant after 950 s in Fig. 9(b), since beacon packets continue to be transmitted even in a disconnected network.

When one flow and two flow modes of TEDD are compared, we observe that one flow minimizes the number of transmissions and two flows maximize the coverage. It can be seen that, when two flows are used, TEDD achieves a 10% greater network coverage [Fig. 9(a)]. On the other side, when one flow mode is used, TEDD sends twice less packets and spends 9% less energy [Fig. 9(b) and 9(c)].

In Fig. 9(c), the *received/transmitted* ratio of all four approaches is shown. It can be seen that TEDD using one flow mode achieves the best result, followed by TEDD using two flows. Flooding-based dissemination presents a ratio equal to one, which was already expected, since every packet that is received by a node is forwarded with probability one. TBF presents the worst *received/transmitted* ratio, due to the overhead of beacon transmission.

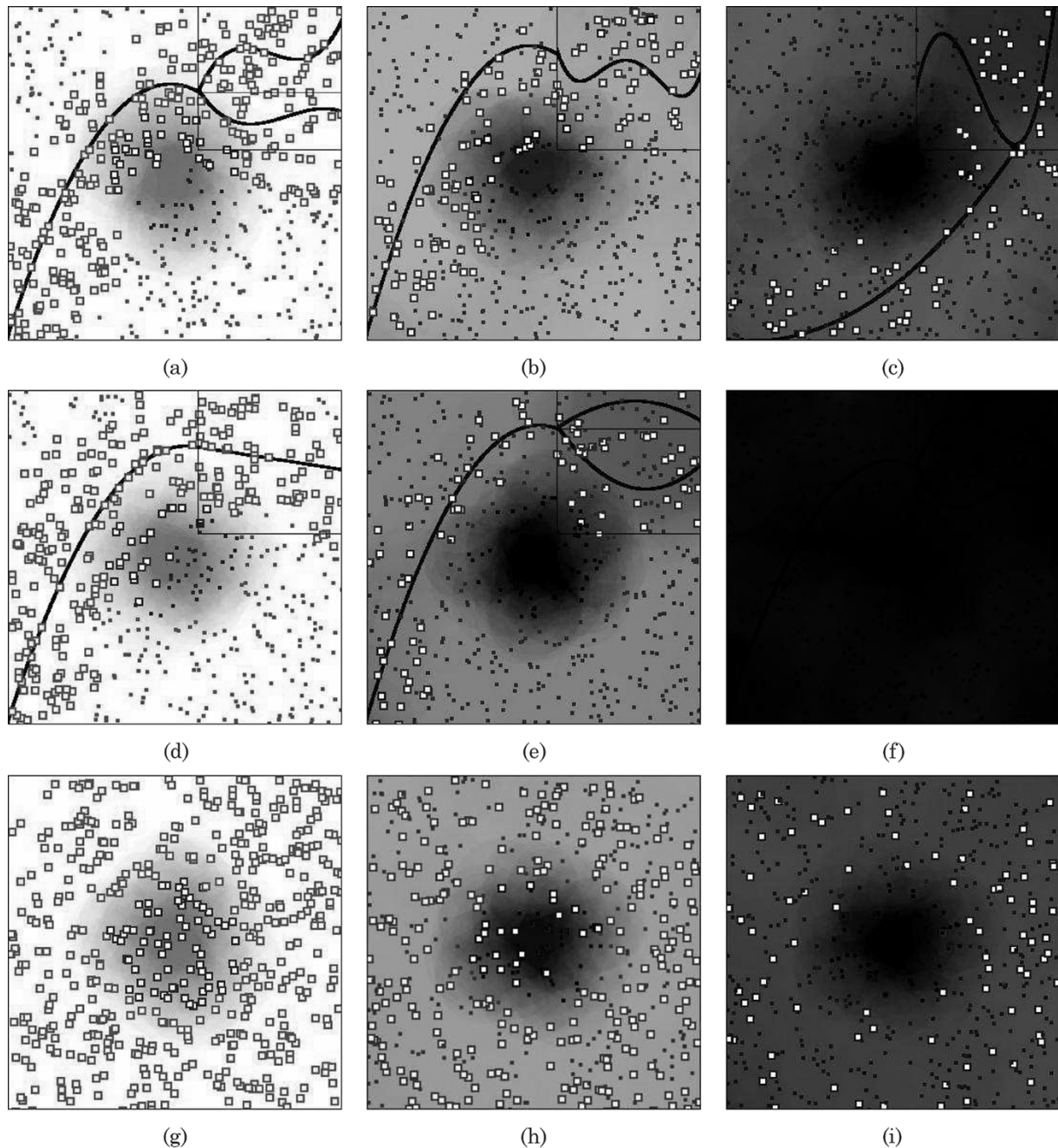


Fig. 10. Energy map and network coverage evolutions for one instance of each protocol evaluated (TBF, TEDD, and *flooding*) in a multicast scenario. (T = time, Ct = coverage inside the target area, $Elea$ = mean energy inside the low-energy area, and En = mean energy in the entire network). (a) TEDD, $T = 0$ s: $Ct = 78\%$, $Elea = 44\%$, $En = 93\%$. (b) TEDD, $T = 500$ s: $Ct = 46\%$, $Elea = 11\%$, $En = 58\%$. (c) TEDD, $T = 1000$ s: $Ct = 36\%$, $Elea = 0\%$, $En = 27\%$. (d) TBF, $T = 0$ s: $Ct = 70\%$, $Elea = 44\%$, $En = 93\%$. (e) TBF, $T = 500$ s: $Ct = 30\%$, $Elea = 0\%$, $En = 43\%$. (f) TBF, $T = 1000$ s: $Ct = 2\%$, $Elea = 0\%$, $En = 2\%$. (g) *Gossiping* dynamic, $T = 0$ s: $Ct = 89\%$, $Elea = 44\%$, $En = 93\%$. (h) *Gossiping* dynamic, $T = 500$ s: $Ct = 34\%$, $Elea = 6\%$, $En = 53\%$. (i) *Gossiping* dynamic, $T = 1000$ s: $Ct = 23\%$, $Elea = 0\%$, $En = 21\%$.

In Fig. 9(f), the latency for TEDD, TBF and flooding are analyzed. The latency is calculated as the time elapsed for each packet sent by the monitoring node until it reaches nodes located at different distances from the monitoring node. It can be seen that TEDD presents a significantly greater latency than the other approaches. This is due to its timing mechanism, which established delays for nodes to forward packets. The delays, as described in Section IV, are used in order to guarantee that only the nearest nodes to the reference points of each dissemination curve forward the packets. This might be the main drawback of

TEDD, what means that it has to be adapted for environments where latency is crucial.

Table II compares the number of operations and radio transmissions performed by TEDD, TBF and flooding. It can be seen that TEDD (one flow) covers less nodes than TEDD (two flows), however, TEDD (one flow) performs significantly less computational operations and transmits twice less packets than the others. Comparing TEDD (both flows) and TBF, the former performs more arithmetic operations, although it realizes less comparisons and assignments, transmits less

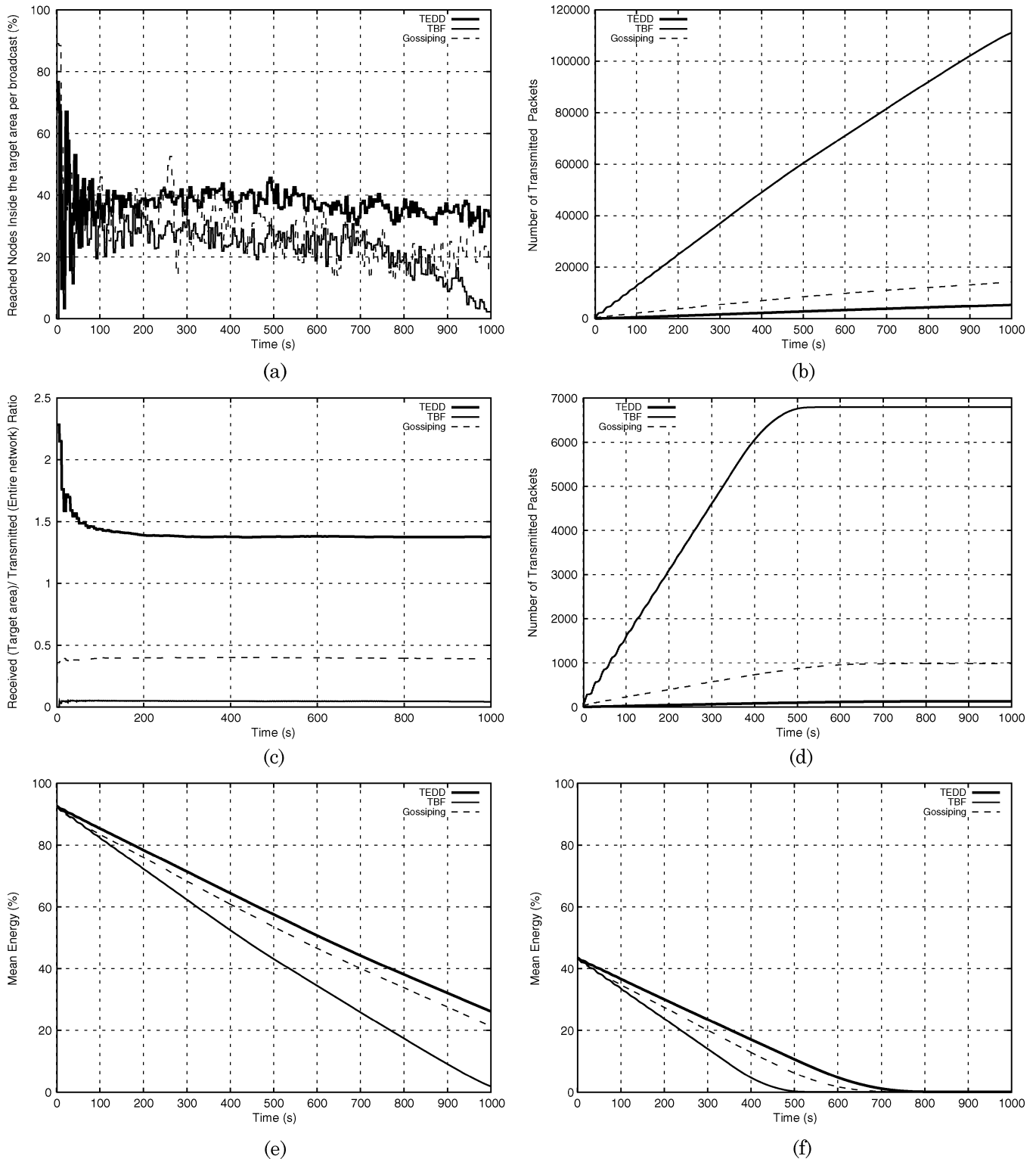


Fig. 11. Performance parameters (TEDD, TBF, and gossiping) in a multicast scenario. (a) Percentage of reached nodes inside the target area. (b) Number of packets transmitted in the entire network. (c) Received (target area)/transmitted (entire network) ratio. (d) Number of packets transmitted inside the low-energy area. (e) Mean energy in the entire network. (f) Mean energy inside the low-energy area.

packets and covers more nodes. Comparing TEDD (both flows) and flooding, the protocol proposed in this work performs more arithmetic operations and assignments, and covers less nodes, however, it transmits less packets and performs less comparisons. This happens because when a node transmits a packet, each one of its neighbors has to process the packet

(e.g., in flooding, each neighbor evaluates whether it has already received the packet). The flooding protocol transmits significantly more packets than TEDD, and performs more comparisons. Finally, considering that the cost of a radio transmission is higher than the cost of a processor operation, TEDD makes an excellent tradeoff.

TABLE III
AVERAGE NUMBER OF OPERATIONS, TRANSMISSIONS AND NETWORK COVERAGE AT EACH DATA DISSEMINATION FOR TEDD, TBF, AND GOSSIPING IN A MULTICAST SCENARIO

Protocols	Operations		Comparisons	Assignments	Tx	Rx
	+	-				
TEDD (both flows)	3.2	3.3	5.9	1.3	27	38%
TBF	1.4	1.2	37.8	22.2	555	25%
Gossiping	—	—	7.3	0.1	71.4	31%

It can be concluded that by avoiding packet transmission by nodes with little energy and establishing trajectories that avoid low-energy nodes, TEDD prolongs the lifetime of these nodes, still guaranteeing that they receive the data disseminated by the monitoring node. When compared with 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, first, the low-energy nodes and, eventually, the entire network. When compared to the TBF forwarding technique, it is important to point out that TEDD is a protocol that does not use neighbor tables, spends much less energy and presents a more adaptive behavior in a dynamic topology scenario.

C. Dissemination to the Target Area

In this section, we analyze a scenario that contains an initial low-energy area and the monitoring node disseminates information to a target area. The low-energy area is located at the middle of the network, and the target area is located at the upper right corner. In this section, we consider three main goals for the data dissemination, all of them having the same relevance: to have the best coverage inside the target area; to transmit the smallest amount of packets in the entire network; and to prolong the lifetime of the nodes located at the low-energy area. The performance of TEDD is compared with both TBF and gossiping,⁸ a flooding-based dissemination scheme with probability [2]. Both TBF and TEDD use the same trajectory generation procedure, described in Section III. Outside the target area, a delivery curve connecting the monitoring 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 are used to forward packets. In Fig. 10(a)–(i), we show the network energy map and the network coverage evolutions during the network lifetime. 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. Moreover, we observe in Fig. 10(a)–(i) the low-energy area and the target area.

Fig. 11(a) illustrates the network coverage inside the target area. We observe that TEDD reaches more nodes, followed by gossiping. TEDD reaches approximately 1.3 times more

nodes than gossiping and 1.5 times more nodes than TBF. In Fig. 11(b), the number of transmitted packets in the entire network is shown. In this case, due to the cost of neighbor table maintenance, TBF sends more packets than the other two approaches. Moreover, gossiping sends 2.6 times more packets than TEDD. Fig. 11(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 1.4, 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 Fig. 11(d), we verify that nodes located inside the low-energy area are not used in the data communication when TEDD is used. Gossiping sends 7.7 times more packets than TEDD, and the TBF sends 53 times more packets than TEDD. The traffic is not completely excluded inside the low-energy area because this region has an intersection with the target area.

A comparison between the energy consumption by the protocols in the entire network and in the low-energy area is shown in Fig. 11(e) and (f), respectively. In both cases, TEDD presents the least consumption and TBF, the greatest. The first result occurs because TEDD has a better selection mechanism of nodes that relay data packets, and the second result is a consequence of the TBF neighbor table maintenance cost. As depicted in Fig. 11(f), TEDD was able to extend the lifetime of nodes inside the low-energy area.

Table III compares the number of operations and radio transmissions performed by TEDD, TBF, and gossiping. Comparing TEDD and TBF, TEDD performs more arithmetic operations, although it realizes less comparisons and assignments, transmits less packets and covers more nodes. Comparing TEDD and gossiping, our protocol performs more arithmetic operations and assignments, however, it realizes less comparisons, transmits less packets, and covers more nodes. TEDD makes an excellent tradeoff between radio transmissions and amount of processing to be done.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed TEDD, a new data dissemination scheme for autonomic WSNs. The key idea is to combine concepts presented in TBF with the information provided by the energy map. We proposed a method for specifying the curves dynamically based on the energy map. We also presented a scheme for dealing with data dissemination to a specific target area. In the original TBF, nodes use a forwarding technique based on neighbors table that consumes more energy. TEDD replaces this

⁸The gossiping protocol works as follows. If a node is outside the target area, the node relays packets with probability of 0.4, otherwise, when a node is inside, it always relays the packets. In this case, the probability is one and the gossiping is equal to flooding.

mechanism with a new forwarding technique: when a node receives a packet, it decides whether it should forward the packet based solely on its own location and the equation embedded in the packet. All these features, when put together, present an autonomous solution to data dissemination in a WSN.

The simulations showed that when TEDD is used, the routing process becomes more adaptive to topology changes. 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, showing in this way the autonomous characteristics of our solution.

There are several improvements that we are planning to introduce. 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. Another future work is to introduce other techniques to avoid transmissions inside the low-energy region. For example, we can use a energy threshold to allow nodes that have less energy than a certain pre-defined amount to not forward data.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, pp. 393–422, 2002.
- [2] Z. Haas, J. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. IEEE 21st Annu. Joint Conf. IEEE Comput. Commun. Soc.*, vol. 3, 2002, pp. 1707–1716.
- [3] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proc. MobiCom*, Seattle, WA, 1999, pp. 174–185.
- [4] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proc. 9th Int. Conf. Arch. Support Programming Languages and Operating Systems*, Nov. 2000, pp. 93–104.
- [5] L. Hughes, O. Banyasad, and E. Hughes, "Cartesian routing," *Comput. Netw.*, vol. 34, pp. 455–466, 2000.
- [6] IBM. (2004) Autonomic computing. [Online]. Available: <http://www.ibm.com/research/autonomic>
- [7] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, Boston, MA, 2000, pp. 56–67.
- [8] D. Johnson, D. A. Maltz, and J. Broch, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, J. Imielinski and J. Korth, Eds. Norwell, MA: Kluwer, 1996, vol. 353.
- [9] J. Kulik, W. R. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," in *Proc. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, Seattle, WA, Aug. 1999, pp. 200–209.
- [10] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Englewood Cliffs, NJ: Prentice-Hall, 1974.
- [11] *MTS/MDA Sensor and Data Acquisition Boards User's Manual*, 2004. www.xbow.com.
- [12] R. A. F. Mini, M. do Val Machado, A. A. F. Loureiro, and B. Nath, "Prediction-based energy map for wireless sensor networks," *Ad Hoc Netw. J.*, vol. 3, pp. 235–253, 2005.
- [13] R. A. F. Mini, A. A. F. Loureiro, and B. Nath, "A state-based energy dissipation model for wireless sensor networks," in *Proc. 10th IEEE Int. Conf. Emerging Technol. Factory Autom.*, 2005.
- [14] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. New York: Wiley, 2001.
- [15] D. Niculescu and B. Nath, "Trajectory-based forwarding and its applications," in *MobiCom*, 2003, pp. 260–272.
- [16] (2002) ns2. The Network Simulator. [Online]. Available: www.isi.edu/nsnam/ns
- [17] C. C. Paige and M. A. Saunders, "LSQR: Sparse linear equations and least squares problems," *ACM Trans. Math. Softw.*, vol. 8, pp. 195–209, 1982.



Max do Val Machado received the B.S. degree in computer science from the Pontifical Catholic University of Minas Gerais, Belo Horizonte, Brazil, in 2002, and the M.S. degree in computer science from the Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2005. Currently, he is working towards the Ph.D. degree in computer science at UFMG.

His research interests are routing and cross-layer design in wireless sensor networks and mobile ad hoc networks.



Olga Goussevskaia received the M.S. degree in computer science from the Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, in 2005.

Her current research interests are primarily in routing and control protocols for wireless sensor networks. Her other interests include optimization models and algorithms for mobile telecommunication systems.



Raquel A. F. Mini received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil.

Currently, she is an Associate Professor of Computer Science at the Pontifical Catholic University of Minas Gerais, Belo Horizonte, Brazil. Her main research areas are sensor networks, distributed algorithms, and mobile computing.



Cristiano G. Rezende received the B.Sc. degree in computer science from the Federal University of Minas Gerais, Belo Horizonte, Brazil.

His research areas are sensor networks and mobile computing.



Antonio A. F. Loureiro received the B.Sc. and M.Sc. degrees in computer science from the Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, and the Ph.D. degree in computer science from the University of British Columbia, Vancouver, BC, Canada.

Currently, he is an Associate Professor of Computer Science at UFMG. His main research areas are wireless sensor networks, mobile computing, distributed algorithms, and network management.



Geraldo Robson Mateus received the M.S. and Ph.D. degrees in computer science from the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, in 1980 and 1986, respectively.

He is a Full Professor of Computer Science at the Federal University of Minas Gerais, Belo Horizonte, Brazil. He spent 1991 and 1992 at the University of Ottawa, Ottawa, ON, Canada, as a Visiting Researcher. He has published over 100 scientific papers and is a leader of several national and international projects. His research interests span network

optimization, combinatorial optimization, algorithms, and telecommunications.

Dr. Mateus is a member of the Institute for Operations Research and the Management Sciences (INFORMS), the International Federation of Operational Research Societies (IFORS), and the Society for Industrial and Applied Mathematics (SIAM).



José Marcos S. Nogueira (M'94) received the B.S. degree in electrical engineering and the M.S. degree in computer science from the Federal University of Minas Gerais (UFMG), Belo Horizonte, Brazil, in 1979, and the Ph.D. degree in electrical engineering from the University of Campinas, Campinas, Brazil, in 1985.

He is an Associate Professor of Computer Science at UFMG. He held a Postdoctoral position at the University of British Columbia, Vancouver, BC, Canada (1988–1989), and currently is on a sabbatical year at

the universities of Evry and Paris VI/LIP 6. He headed the Department of Computer Science, UFMG from 1998 to 2000. Currently, he heads the Computer Network Group, UFMG. He has supervised a number of Ph.D. and Master's students. He was the Technical Coordinator of the System for the Integration of Supervision (SIS) Project, where a complex and distributed system for the management of telecommunications networks was developed. He has served in various roles, including General Chair (1985) and TPC Chair (1999 and 2004) of the Brazilian Symposium on Computer Networks (SBRC), and General Chair of LANOMS 2001. He publishes regularly in international conferences and journals. His areas of interest and research include computer networks, wireless sensor networks, telecommunications and network management, and software development.

Dr. Nogueira has been a TPC member in IEEE/IFIP NOMS (2000, 2002, and 2004), IEEE/IFIP IM 2003, IEEE LANOMS (1999, 2001, 2003, and 2005), IEEE/IFIP MMNS (2000, 2001, 2002, and 2003), IPOM 2002, SBRC (from 1990 to 2005), and IEEE/IFIP DSOM (2003, 2004, and 2005). He is a member of the Brazilian Computer Society (SBC) and Brazilian Telecommunications Society (SBTr). He also participates in the IEEE ComSoc CNOM Interest Group. Currently, he is a Secretary of the IEEE ComSoc TCII.