

# DISAGREE: disagreement-based querying in wireless sensor networks

Martha Montes-de-Oca · Javier Gomez · Miguel Lopez-Guerrero

Published online: 30 August 2013  
© Springer Science+Business Media New York 2013

**Abstract** Traditional data acquisition methods for wireless sensor networks (WSNs) require all sensor nodes to transmit at least once to the sink node to obtain a full view of the network. In this paper we present Disagree, a different data acquisition method to retrieve data in WSNs that works in the opposite way as compared with traditional methods. In Disagree, only nodes that do not satisfy an assertion are required to transmit data back to the sink node in order to obtain a complete view of the network. We show that this behavior is the base of an energy-efficient way to gather all data. An important feature of Disagree is that it saves energy at the sensor level by exploiting data correlation. Rather than requesting explicit data readings from all sensor nodes, Disagree estimates the readings from sensor nodes that did not respond to the assertion. As a result of this policy, Disagree can obtain a view of the sensing field with different levels of resolution involving the transmission of only a subset of the sensor nodes by exploiting spatial data redundancy. We implemented Disagree in NS-2 network simulator and results indicate Disagree can significantly reduce the percentage of nodes replying to queries compared with flat and a cluster head based approaches.

**Keywords** Wireless sensor networks · Energy-savings · Data acquisition

## 1 Introduction

A wireless sensor network (WSN) consists of tiny wireless devices known as *sensor nodes*. Each sensor node acquires information about a physical phenomenon, and readings from various sensors are usually transmitted to a main and remote node named *sink node* through a multi-hop ad-hoc network. They are usually sent in response to queries from the sink node. Commonly, sensor nodes have limited energy resources, therefore it is necessary to save energy in order to extend network lifetime.

Energy-efficient data acquisition has been an important issue since the beginning of WSNs, and many researchers have approached this problem in various ways [2, 3]. In general, data acquisition methods can be classified according to how the WSN is organized. One of such classifications divides the data acquisition methods into *flat, hierarchical and dominating set categories* [12].

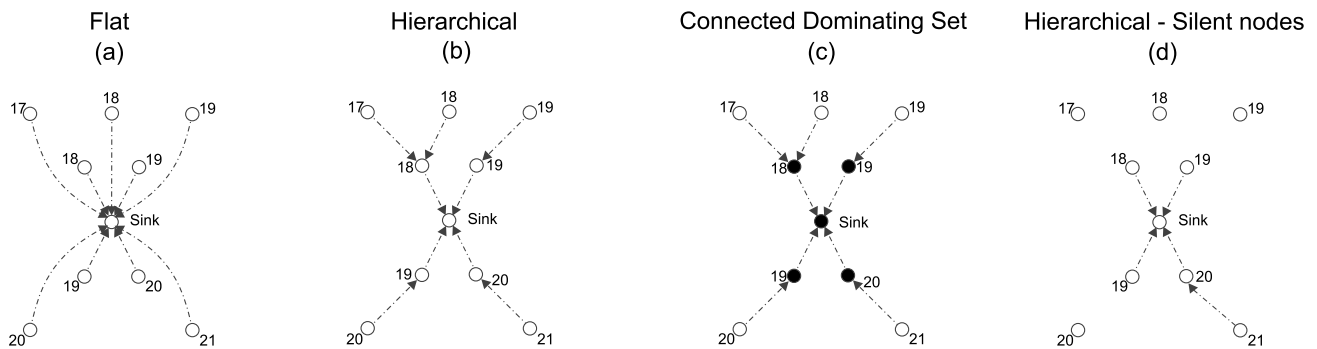
In flat networks [1–3], each sensor will individually transmit its reading to the sink node through a multi-hop route, even if there is a small difference among neighboring readings, which is clearly inefficient in terms of the total number of transmitted packets. Hierarchical [2, 5] approaches reduce the amount of data sent to the sink node by organizing the network into clusters. For each cluster there is a node called cluster head which aggregates individual sensor readings into a single packet that will be sent to the sink node. In general, each sensor becomes associated to a specific cluster head. Hierarchical approaches still require individual sensors to respond to the query sent by the sink node, thus using valuable energy at sensor nodes.

---

M. Montes-de-Oca (✉) · J. Gomez  
Department of Telecommunications Engineering, National  
Autonomous University of Mexico, Mexico City 04510, Mexico  
e-mail: montesdeocacaliz@gmail.com

J. Gomez  
e-mail: javierg@fi-b.unam.mx

M. Lopez-Guerrero  
Department of Electrical Engineering, Metropolitan Autonomous  
University, Mexico City 09340, Mexico  
e-mail: milo@xanum.uam.mx



**Fig. 1** Approaches to data acquisition in WSN

Dominating set approaches [12] organize the sensor network using a connected dominating set (CDS) to form the backbone. Nodes that are not part of the dominating set are adjacent to at least one node that belongs to such a set. In some solutions based on this scheme, members of the dominating set are cluster heads which aggregate readings from sensor nodes. Flat, hierarchical and dominating set methods usually generate as many responses as sensors in the network area are when a full view of the sensing field is needed. If there is little variability among sensor readings we can infer that there should be a more efficient way to get a full view of network involving fewer sensor transmissions.

Disagree, like other proposals, attempts to reduce the amount of data sent to the sink node, however, what is important about our protocol is that we achieve this goal by limiting transmissions at the sensor node level. That is, Disagree does not require all sensors to respond to a query sent from the sink node even if a full view of the network is requested, yet the sink node can estimate readings from silent nodes within some predefined error margin. Therefore, sensors can save energy thus increasing network lifetime.

Disagree organizes the network using a tree structure and there are two kind of nodes: *Tree nodes* are nodes that belong to the distribution tree and *Sensor nodes* which do not belong to the tree but are associated to *Tree nodes*. Sensor nodes receive assertions and transmit their readings only if they *disagree* with the assertion. We will show that this is the key idea of an energy-efficient way to retrieve data in WSNs. Because in many sensing applications there is spatial correlation of data readings among sensors located nearby, a common situation in Disagree is that only a few sensor nodes will have readings too different from the one reported by the tree node (in Disagree a tree node is similar to a cluster head). This allows many sensor nodes to remain silent, thus saving valuable resources such as energy and bandwidth.

Rather than requesting responses from nodes having readings within a fixed interval as in traditional methods, Disagree uses a dynamic interval centered at the value of the reading reported by the corresponding tree node with a tolerance above and below defined by a parameter named

delta ( $\Delta$ ). Sensors having readings located outside this interval *disagree* with the tree node's assertion by responding to it. The delta value in Disagree can be used to increase or decrease the resolution of sensed readings at sensor nodes. Larger values of delta provide low spatial resolution since more nodes remain silent. On the other hand, smaller deltas provide higher spatial resolution as more sensors disagree by transmitting their reading to their associated tree node.

Disagree's operation leads to varying degrees of accuracy during the estimation of readings from silent nodes whenever delta is set to a value different of 0. However, because delta is a parameter controlling the accuracy of estimated readings in the entire network; it is up to the application to decide which value of delta fits well with its intended purpose/operation. For instance, an application sensing temperature over an airport might tolerate an error of 3–5 degrees, while an application sensing temperatures inside a reactor might require estimation errors to be below 1 degree. By controlling the value of delta, Disagree can also be configured as a multi-resolution data acquisition protocol. For instance, Disagree could initially request a low resolution view of the network using a large delta value, then a second query may be propagated in specific regions of the network using a small delta value to obtain more spatial detail.

Figure 1 illustrates the basic operation of four acquisition methods after a sink node requests a full view of network. The numbers in Fig. 1 indicate examples of the readings at each node. Figure 1a shows a flat topology where all nodes send their reading to the sink node individually, Fig. 1b shows a hierarchical approach where nodes send their reading to cluster heads, which then send the aggregate data to the sink node. In Fig. 1c, nodes forming a connected dominating set (CDS) are shown as black nodes. Sensor nodes (shown as white nodes) send their readings to their nearest neighbor in the CDS which aggregates and forwards data to the sink node. This case is very similar to a hierarchical approach. Finally, Fig. 1d shows a hierarchical approach where there are silent nodes that do not transmit their readings to the sink node but it is possible to estimate readings from silent nodes within an error margin. As we can

see, Disagree belongs to this last category. As we will show later, with Disagree it is easier to obtain data with different spatial resolution and its performance improves as the spatial correlation of readings increases, that is, as deviations of data readings from the central value reported by the tree node become smaller, which is a common property of many physical phenomena. Similarly, the performance of Disagree increases as more nodes are deployed in the network. We argue that various WSN applications need spatially dense sensor deployments in order to achieve satisfactory coverage and spatial resolution. However, even in low density situations Disagree performs well compared with traditional hierarchical techniques as we will show in the performance evaluation section.

As we will show later, the performance of Disagree improves as the spatial correlation of the readings increases, that is, as deviations of data readings from the central value reported by the tree node become smaller, which is a common property of many physical phenomena

The rest of the paper is organized as follows. Section 2 reviews some related work in this area. Section 3 details the operation of Disagree while Sect. 4 describes a model for predicting the number of silent nodes for a given delta value. Section 5 describes the implementation of Disagree, experiments and obtained results. Finally, in Sect. 6, we provide some final remarks.

## 2 Related work

As we mentioned before, data acquisition methods for WSNs can be classified into flat, hierarchical and dominating set categories, according to how the network is organized.

In flat networks, each sensor plays the same role and is assumed to be equipped with the same resources as everyone else. One of the pioneer data acquisition methods reported in the literature using a flat network structure was Directed Diffusion [19], in which queries are flooded in the network and sensor nodes reply to such queries through reverse paths. The emphasis of directed diffusion was mainly on how to route collected data back to the sink node, rather than optimizing energy consumption at sensor nodes. Other approaches focused mostly on improving the query propagation phase of data acquisition methods. Traditional flooding techniques commonly used to propagate queries require each node to broadcast a query packet once, generating many redundant transmissions and consuming valuable energy resources [30]. In Gossiping [17], nodes do not re-broadcast every query message they receive, but rather they do so randomly. In Rumor routing [6], sensors create paths to events so that a query can follow these paths rather than flooding the entire network with query packets. This operation saves energy during the query process, however it needs

energy to create these paths, so the final energy savings will depend on the specific sensing application and network characteristics.

Hierarchical methods use a node called cluster head, which collects individual readings from associated sensor nodes, then aggregates collected information into a single response that will be sent to the sink node. In general, each sensor becomes associated to a specific cluster head, and cluster heads may form multiple levels of hierarchy. Because only a single response in a region is routed to the sink node, as opposed to many individual responses, energy is saved. There are various examples of cluster head approaches in WSNs. In Low-Energy Adaptive Clustering Hierarchy (LEACH) [7], cluster heads collect data locally using TDMA and rotate the cluster head role among sensors in the cluster to avoid energy depletion. In [24] the authors propose an algorithm called Density and Distance based Cluster Head Selection (DDCHS) which dynamically selects as the new cluster head the node being the nearest to the previous cluster head having the highest number of neighbors. In [39] the authors propose a Hybrid Energy Efficient Distributed (HEED) where cluster head selection is primarily based on the residual energy of each node in order to balance energy consumption and all nodes are assumed to be equipped with the same initial energy. In H-HEED [23] the authors consider the heterogeneity in terms of node energy while rotating the cluster head role. In [21] the authors propose an energy-efficient data collection protocol based on a tree (EEDCP-TB). This method chooses nodes having more residual energy as forwarding nodes in order to balance energy consumption. In [20] the authors propose a load balancing algorithm to reduce energy consumption for heterogeneous WSNs. In [15] the authors propose DGLB which classifies sensor nodes into different layers based on hop counts between the sink node and sensor nodes. This algorithm provides a uniform energy consumption among sensor nodes. A common feature of all of these methods is that they rotate the cluster head role but they do not exploit data correlation (spatial or temporal) because all nodes send their readings to the local cluster head, which then forwards such readings to the sink node.

Hierarchical methods exploiting temporal correlation include Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [28] which is similar to LEACH, except that the cluster head provides two thresholds (hard and soft). In case the current sensor reading is greater than the hard threshold and the difference between the current and previous readings is greater or equal than the soft threshold, the sensor node transmits its new reading to the cluster head. TEEN is oriented to reactive sensing applications and does not perform well in proactive applications where periodic reports are needed. Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol (APTEEN) [29],

combines TEEN with TDMA functionality to provide better performance for both proactive and reactive spatial sensing applications. In [4] the authors propose a protocol called SHRP whose main goal is to find a route that increases network lifetime exploiting temporal correlation in sensor nodes. In this work, each sensor node verifies how much the latest sensed value has changed with respect to the last value to decide whether or not to send the new value to the cluster head. At this point it is worth mentioning that we do not compare Disagree with any of these methods because Disagree exploits spatial correlation.

Hierarchical methods exploiting spatial and temporal correlation in WSNs generally use a hierarchical approach in which cluster heads apply compression techniques to collected data readings in order to reduce the size of the response. These methods may consider a multi-resolution scheme in which the sink node may establish the desired spatial resolution. In general, higher resolution implies higher energy expenditure as more data need to be sent to the sink node. In Multi Resolution Compression and Query framework (MRCQ) [36], the authors use a hierarchical approach where data reported from a lower layer are compressed by an upper layer node using spatial coding techniques. In this method only low resolution views of the network are reported to the sink node in order to save energy. However, high resolution information is kept in upper layer nodes of the hierarchy to allow for occasional high resolution queries. DIMENSIONS [14] uses wavelets to compress long-term data traces, and this method can be used to retrieve higher or lower resolution views of the network, if necessary. We emphasize that unlike these methods Disagree does not use compression techniques to reduce the number of transmitted packets.

Protocols suited to event detection in the network usually exploit temporal correlation. In [9] the authors propose a method called Ken to save energy at the sensor level. In this work sensor nodes use a probabilistic model to predict the next reading based on previous readings. When the new reading is different from the expected reading, sensor nodes proactively send the new reading toward the sink node. In [34, 35], only members of a representative set of nodes communicate their readings to the sink node in order to meet a distortion constraint. These methods, however, are sensitive to data variations in time so they require all nodes to periodically send their readings to the cluster head or to the sink node to update the query tree. Additionally, these methods provide a single spatial resolution of sensed values in the network. Opposite to these protocols, Disagree is not sensitive to variations in data correlation in time so the query tree does not need to be updated. Similarly, Disagree can provide views of the network with different spatial resolution depending of the particular application being used.

In the hierarchical—silent nodes category, where Disagree belongs, there are various protocols. SMRDD [8] divides a sensor field into many small regions called grids, where the grid size corresponds to the desired data resolution level. In this method there are two kinds of nodes; pixel points that are nodes similar to cluster heads and home nodes which are associated to pixel points. Each grid has one pixel point and one or more home nodes. Queries are sent to pixel points along paths in a tree composed of the sink node and all pixel points. SMRDD cannot retrieve or estimate readings from sensors that do not belong to a particular grid size, thus collecting much less information compared with Disagree. In order to compare SMRDD to Disagree, SMRDD would need to obtain or estimate readings from all sensors, this would need a tree involving all sensors, and this is exactly the same as a cluster head approach. In CAG [38], the authors retrieve data in a WSN using a subset of representative nodes in the network. CAG operates in two phases named query and response. During the first phase, a query is disseminated through the network using a tree composed of cluster heads and bridge nodes. Based on a cluster head reading ( $CR$ ), a node's reading ( $MR$ ) and a threshold ( $\tau$ ), a node decides to join a cluster only if  $MR < CR \pm CR \times \tau$ . Otherwise, the node declares itself as a new cluster head, sending a new broadcast query to neighboring nodes with its own reading. During the response phase, only cluster heads transmit their readings back to the sink node. In [18, 22] each node overhears readings from its neighbors in order to determine the redundancy of its own reading. This method exploits spatial data correlation to decrease the number of transmitted data packets, but frequently, all nodes should transmit control messages in order to determine the redundancy of new readings.

Dominating set approaches use a CDS as backbone of the network. In [16] the authors propose an algorithm to construct a CDS based on data spatial correlation. This protocol has an initial phase where all nodes transmit their readings to their neighbors in order to create the CDS. Afterwards, each node tests itself if its current reading can be inferred as a linear combination of readings from neighboring nodes. If this condition is satisfied the node informs other nodes that it will not be responding to future queries. During the second phase only nodes in the CDS send their readings to the sink node in order to decrease the number of transmitted packets. This protocol has a maintenance phase in order to construct a new CDS suited to reading changes. This method cannot be used to obtain multiresolution because only nodes in the CDS transmit their readings and only one resolution is reached. Because this work cannot be used to obtain multiresolution and needs a periodic maintenance phase where all nodes transmit, it is not directly comparable with Disagree. Similar situation happens in [11], where the authors propose to select a few nodes to form a virtual backbone in order to support routing and monitoring.



As we summarized, most hierarchical and multi-resolution methods proposed for WSNs exploit data redundancy at the cluster head level, requiring individual sensor nodes to transmit their readings to their associated cluster head. The key difference with these techniques is that Disagree exploits spatial data redundancy down to the sensor level, estimating readings from silent nodes, thus allowing sensor nodes to avoid costly data transmissions when unnecessary.

To assess the relative performance of Disagree, we selected LEACH because it is an example of a hierarchical approach that includes a cluster-head rotation algorithm and, in this particular point, is similar to Disagree. We also selected CAG because this protocol, in a similar way to Disagree, exploits spatial data redundancy at the sensor level by allowing some nodes to remain silent and it is able to infer readings from silent nodes. However, it is worth emphasizing that CAG periodically forces all sensor nodes to transmit in order to update its query tree. In this aspect it differs from our proposal. In spite of this difference and to the best of our abilities, we believe that CAG is the closest in operation to Disagree. We did not compare Disagree with more protocols because we did not find other proposals that share some other key characteristics with Disagree. The results of comparing Disagree against LEACH and CAG are presented in Sect. 5.

### 3 Disagreement-based sensing (Disagree)

Disagree is optimized for proactive [28] sensing applications which require a full view of the network. The operation of Disagree consists of three main phases namely expansion, disagreement and contraction. There is also an initial phase that occurs only once at the beginning of the operation of Disagree when the backbone is constructed using a minimum spanning tree algorithm [10]. In order to avoid the premature death of some tree nodes, a fourth phase rotates the tree role among nodes balancing energy consumption in the network. These phases are detailed below.

#### 3.1 Tree creation

Disagree organizes the network using a tree structure [10] as in a hierarchical approach. Within this structure, there are two kinds of nodes: *Tree nodes* ( $TN$ ) are nodes that belong to the distribution tree, with the special case of  $TN_0$  which is the root of the tree or *Sink node*, and *Sensor nodes* ( $SN$ ), which do not belong to the tree.

In order to construct the tree structure, Disagree uses Prim's minimum spanning tree (MST) algorithm [10]. This algorithm considers the maximum distance between neighbor nodes as the minimum weight. The construction of tree structure is centralized at the sink node knowing the location of all nodes. For simplicity, we assume that each

node knows its location (using GPS or any another location method [33]).

At the beginning of the algorithm, the sink node places itself as the root of the tree. For each step, the algorithm adds to the tree a new node having the maximum distance considering the distances from nodes in the tree and their neighbors. This procedure continues until all nodes are included in the tree.

Once the MST is created, the sink node determines if a node is  $TN$  or  $SN$  considering that  $TNs$  may or may not have  $SNs$  associated to them, but every  $SN$  must be associated to a unique  $TN$ . Finally, the sink node sends to each node a packet to indicate if the node is a  $TN$  or a  $SN$ , which  $TN$  is its parent node and which  $SNs$  are its children nodes.

It is important to mention that other protocols such as CAG [38] select tree nodes based on readings. However, a problem with this approach is that whenever the phenomenon being monitored changes over time it makes necessary to update the query tree periodically. Updating the query tree is a costly process generating additional signaling packets and their associated energy consumption. In Disagree, on the other hand, we used distances to select tree nodes precisely to avoid updating the query tree every time nodes sensed a different reading.

#### 3.2 Expansion

After a minimum tree is set up, the sink broadcasts an assertion packet along the tree to cover the entire network. This assertion packet has the following structure:  $Assertion(METRIC, READING, \Delta)$ , where  $METRIC$  is the metric being monitored (temperature, humidity, etc.),  $READING$  is the reading of the  $TN$  forwarding the assertion, and  $\Delta$  is used to establish a range. Upon receiving an assertion packet, a  $TN$  updates this packet with its own reading before forwarding it.  $SNs$  receiving an assertion packet do not retransmit it. For example,  $Assertion(temp, 28, 3)$  (represented as  $A(temp, 28, 3)$  for short) in Fig. 2 means that a  $TN$  claims that all its associated  $SNs$  have a temperature reading inside the range  $[28 \pm 3]$ . Before retransmitting an assertion, every  $TN$  updates the assertion with its own reading of the variable being monitored, see Fig. 2. In this figure we can observe that each  $TN$  updates its own assertion with its reading before retransmit it. For instance  $TN_A$  sends  $Assertion(temp, 30, 3)$  after receiving  $Assertion(temp, 28, 3)$  from the sink node.

#### 3.3 Disagreement phase

This phase is the most important part of Disagree. To explain the rules governing the decisions taken by  $SNs$  after they hear an assertion, we refer to Fig. 3. As we explained before, only sensors having readings outside the reported range

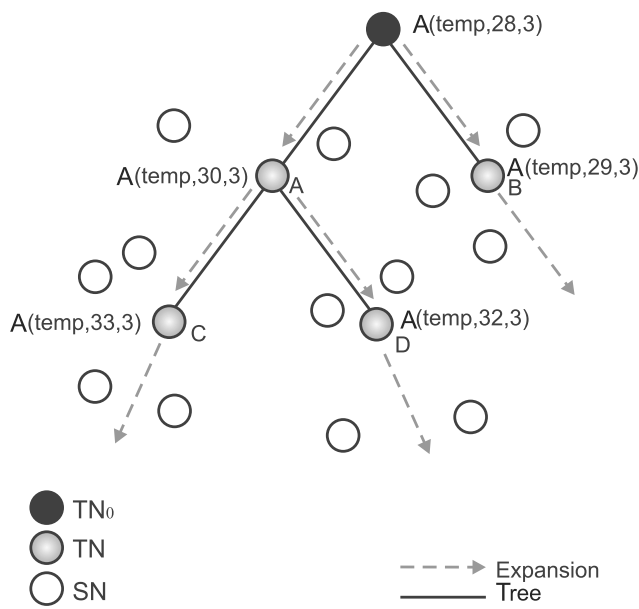


Fig. 2 Tree creation and expansion phases

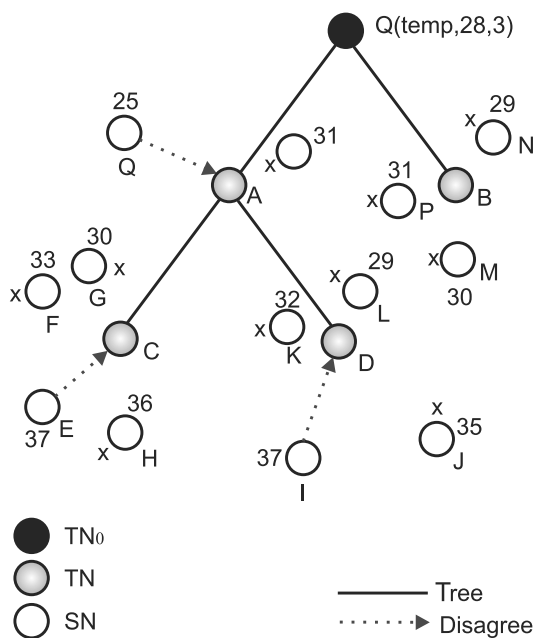


Fig. 3 Disagreement phase

$[temp \pm 3]$  in the example depicted in Fig. 3 must disagree transmitting their reading to their associated  $TN$ . In Fig. 3  $TNs$  and  $SNs$  are denoted with letters and the readings of each  $SN$  are indicated with numbers.

In the example of Fig. 3 only sensors  $SN_E$ ,  $SN_I$  and  $SN_Q$  transmit their readings to their associated  $TN$  using unicast packets (dashed line). All others  $SNs$  (marked with an  $x$  in the figure) remained silent. Even though some sensors did not reply to their associated  $TN$ , it is easy for a

$TN$  to infer that readings from silent nodes are somewhere within the range  $[temp \pm 3]$ .

For cases where data deviations from the reading reported by a  $TN$  are small, it is expected that only a few  $SNs$  will disagree with the assertion, thus permitting the majority of sensor nodes to remain silent. On the other hand, if data deviations around the reading reported by a  $TN$  are large, or the delta value is small, almost all associated  $SNs$  will disagree with the assertion forcing them to reply.

Since the operation of Disagree may cause some sensors to remain silent, it is necessary to distinguish normal silent nodes from nodes that no longer have energy at their disposal. In order to achieve this, we recommend that each sensor node to periodically send a hello packet to its associated tree node. Once a  $TN$  detects that a node is no longer alive, it sends a notification back to the sink node.

### 3.4 Contraction

During the contraction phase, collected information at  $TNs$  is transmitted back to the sink node, see Fig. 4. This phase is initiated by  $TNs$  located in the boundary of the sensor network after they finish collecting disagreements from associated  $SNs$ .  $TNs$  can discover that they are located at the boundary of the network if no other  $TN$  is associated with them. Because data aggregation is not the key contribution of Disagree, in this paper we did not consider an aggregation/compression method to reduce the size of collected data at  $TNs$ . However, if necessary, some methods exploiting spatial and temporal redundancy techniques [13, 25–27, 31, 32, 37] can be used without affecting the basic behavior of Disagree. In our implementation,  $TNs$  aggregate collected data simply by concatenating the ID and reading of disagreeing  $SNs$ , adding its own ID and reading into a single packet that is sent towards the sink node. In order to keep to a minimum the number of transmitted packets in this phase, when a  $TN$  has two or more upstream  $TNs$  associated to it, the  $TN$  waits until it hears from each and every upstream  $TN$ , see Fig. 4. Once the information is concatenated in a tree node, it is routed back to the sink node.

Upon collecting all data from  $TNs$ , the sink node can reconstruct a full view of the network as depicted in Fig. 5. In this figure, we show an example in which the sink node collects data from  $TN_A$ ,  $TN_B$ ,  $TN_C$ , and  $TN_D$ . With this information the sink node can reconstruct a full view of the sensor network. As we can see in this example, only  $SN_E$ ,  $SN_I$  and  $SN_Q$  disagreed sending their readings to their associated  $TN$ . For all other nodes in the network, the sink node only has an estimated value of their readings.

Disagree operation is shown in Algorithm 1. The computational complexity of the proposed algorithm is  $O(2N)$ .

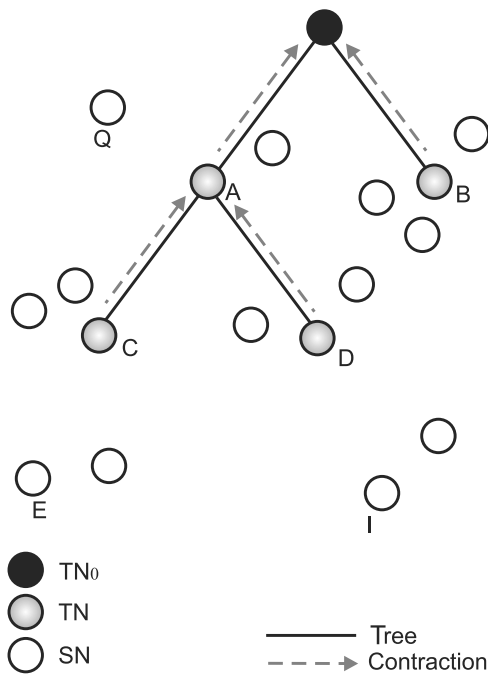


Fig. 4 Contraction phase

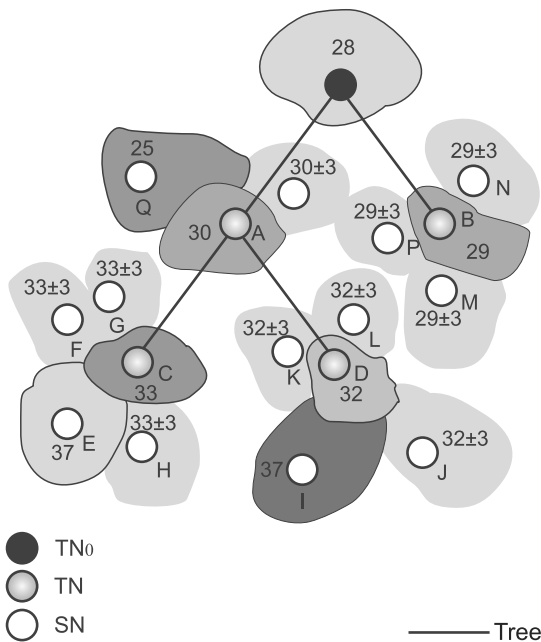


Fig. 5 Reconstructing a full view of the network

The complexity of Prim’s algorithm to create the initial tree is  $O(E \log N)$ , where  $E$  is the number of edges and  $N$  is the number of nodes.

### 3.5 Tree rotation

In the operation of Disagree, tree nodes are likely to transmit more often than sensor nodes, thus consuming more en-

### Algorithm 1 Disagree( $N, TN, SN, S$ )

```

/*N: all nodes*/
/*TN: nodes in tree*/
/*SN: nodes not in tree*/
/*S: sink node*/

1: S broadcasts an assertion packet ( $pkt_a$ )
2: for  $i = 1 \rightarrow N$  do
3:   if  $N_i \in TN$  then
4:     Receives  $pkt_a$  from its father  $TN_i$ 
5:      $fathervalue = value$ 
6:     Updates  $pkt_a$  with its own reading ( $value_{N_i}, \delta$ )
7:     Broadcasts  $pkt_a$ 
8:   else
9:     Receives  $pkt_a$  from its father  $TN_i$ 
10:  end if
11: end for
12: for  $i = 1 \rightarrow N$  do
13:  if  $N_i \in SN$  then
14:     $value = \text{sensing the phenomenon}$ 
15:    if  $value > fathervalue + \delta$  or  $value < fathervalue - \delta$  then
16:      Transmits its  $value$  to its father  $TN_i$ 
17:    else
18:      Do not transmit its  $value$ 
19:    end if
20:  else {  $N_i \in TN$  }
21:    if  $TN_i$  has child(s) then
22:      Waits certain time for readings
23:      Transmits all received readings to sink
24:    else {  $N_i$  has a child node  $\in TN$  }
25:      Waits until receives reading from its  $TN_i$  child
26:      Transmits all received readings to sink
27:    end if
28:  end if
29: end for
    
```

ergy. Tree node activities include forwarding assertions, collecting and aggregating disagreements from associated sensor nodes during the disagreement phase, and forwarding of collected data packets from other tree nodes to the sink node during the contraction phase. The premature death of a tree node can be avoided by some methods such as equipping tree nodes with larger batteries, or periodic rotation of the tree node role among nodes in the network. In our implementation we decided to implement a tree rotation algorithm (see Algorithm 2). The rotation algorithm is applied at periodic intervals that we call epochs. All nodes keep a counter of how many epochs have served as tree node. The sink node starts the rotation process by flooding a small control packet called *rotatetree*. Before retransmitting the *rotatetree* packet, each node updates this packet with its

own ID and epoch counter, and keeps a record of all IDs and epoch counters ( $cTN$ ) overheard from neighboring nodes associated to the same  $TN$ . Each node selects the neighbor having the minimum epoch counter as its new tree node. In case two or more nodes have the same minimum epoch counter, a node selects one of them randomly. Finally, a node sends a unicast packet to the selected tree node to establish the association.

Figure 6 illustrates the rotation algorithm. Figure 6a shows a tree from epoch 1, while Fig. 6b shows the resulting tree after applying the rotation algorithm.

Another important issue in Disagree is how to set the value of  $\Delta$ . This decision depends on the needed resolution and allowed error tolerance of each particular applica-

---

**Algorithm 2** Tree rotation( $N, TN, SN, N, epoch, level$ )

---

/\*N: all nodes\*/

/\*TN: nodes in tree\*/

/\*SN: nodes are not in tree\*/

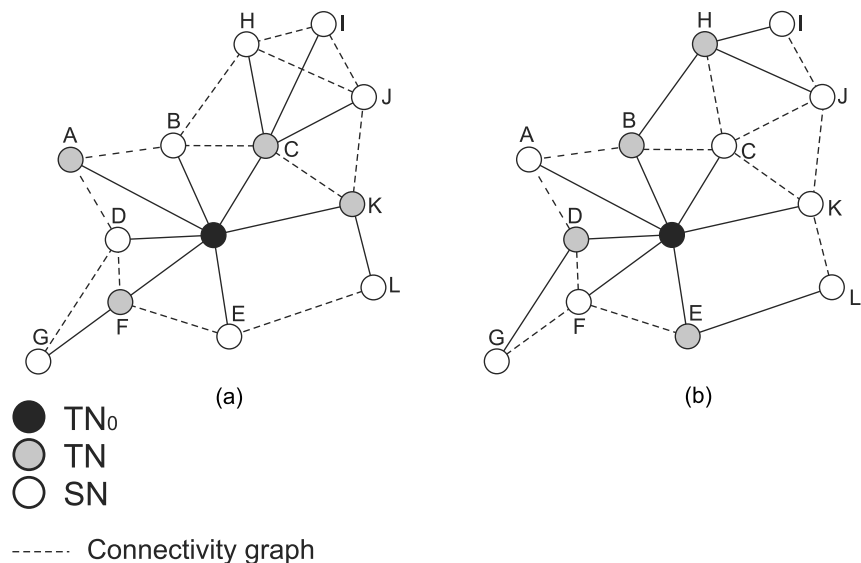
/\*S: sink node\*/

/\*epoch: number of rounds\*/

/\*level: number of levels in tree\*/

- 1: **for**  $i = 1 \rightarrow epoch$  **do**
  - 2:   **for**  $j = 1 \rightarrow level$  **do**
  - 3:      $S$  broadcasts *rotatetree*( $ID, cTN$ )
  - 4:     Each node receives *rotatetree* and saves the information
  - 5:     Each node updates this packet with its own ( $ID, cTN$ ) and broadcasts it
  - 6:     Each  $SN$  should associate with the  $TN$  that has a minimum  $cTN$
  - 7:   **end for**
  - 8: **end for**
- 

**Fig. 6** Tree rotation. (a) Shows a tree from epoch 1 and (b) shows the resulting tree after applying the rotation algorithm for epoch 2



tion. Finally, Disagree is a protocol initially thought to exploit spatial data correlation, however, this protocol could be adapted to take advantage of temporal correlation also.

#### 4 Exploiting data redundancy in Disagree

In this section we explain in more detail how Disagree is able to exploit spatial redundancy in WSNs. We also describe a simple method that can be used to compute whether or not a sensor node will disagree to an assertion; and we will discuss how this method can also be used to compute energy savings for our protocol when compared to a traditional cluster-head approach.

Let us consider a scenario as the one depicted in Fig. 7 top, where a static sensor network monitors a certain variable. In this figure the surface represents the value of the metric being monitored at each point in the network.

As an example let us focus on  $TN_3$  which, after receiving an assertion and updating the  $READING$  field with its own reading, broadcasts such assertion as  $Assertion(METRIC, READING_{TN_3}, \Delta)$ . Now let the curve  $Reading(x, y) = Z_0$  represents a level curve of the surface having the same height  $Z_0$ .

We are interested in the following two level curves for  $TN_3$

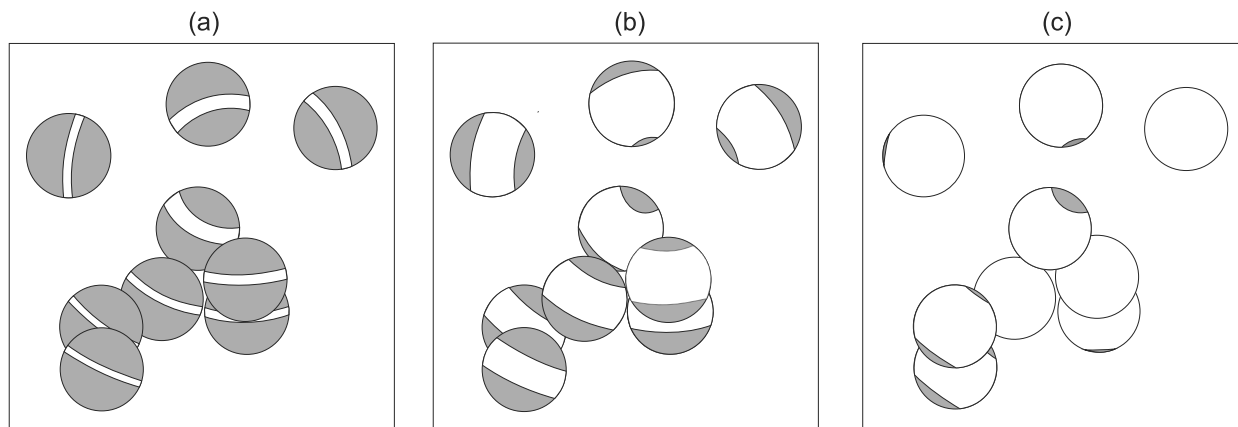
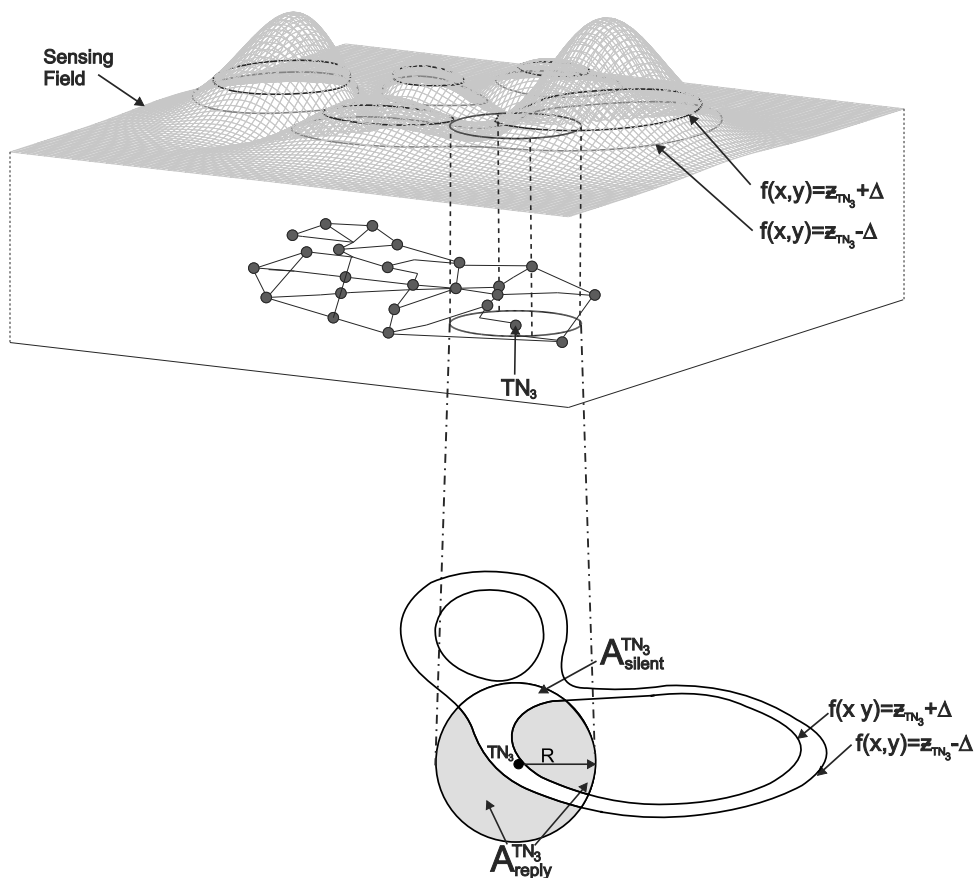
$$f(x, y) = READING_{TN_3} + \Delta \quad (1)$$

$$f(x, y) = READING_{TN_3} - \Delta \quad (2)$$

These two equations correspond to level curves of  $Reading(x, y)$  having heights of  $READING_{TN_3} + \Delta$  and  $READING_{TN_3} - \Delta$ , respectively (see Fig. 7 top). Again, only associated nodes outside the range  $READING_{TN_3} \pm \Delta$



**Fig. 7** (Top) A sensor network on a sensing field. The surface represents the values of the metric in the network where two level curves are indicated ( $f(x, y) = Z_{TN_3} + \Delta$  and  $f(x, y) = Z_{TN_3} - \Delta$ ). (Bottom) We focus on the  $TN_3$  and zoom around it to observe how the two level curves establish the non-gray and gray areas around  $TN_3$



**Fig. 8** Disagreeing areas (gray) for (a)  $\Delta = 1$ , (b)  $\Delta = 5$ , and (c)  $\Delta = 10$

should respond to the assertion while all other associated sensors should remain silent. This situation is illustrated in Fig. 7 bottom, where only associated  $SNs$  located inside the gray area ( $A_{reply}^{TN_3}$ ), enclosed by the coverage range of  $TN_3$  and the two level curves, should respond to the query. Nodes located in non-gray areas ( $A_{silent}^{TN_3}$ ) should remain silent. From the figure we observe that  $A_{silent}^{TN_3} = A_{coverage}^{TN_3} - A_{reply}^{TN_3}$ , where  $A_{coverage}^{TN_3} = \pi R^2$  is the area covered by the transmission range of  $TN_3$  with a radius of  $R$

meters. Similarly, assuming uniform density of nodes, the probability that a  $SN$  associated to  $TN_3$  remains silent after it receives an assertion can be approximated as

$$Probability_{silent} \approx \frac{A_{silent}^{TN_3}}{A_{coverage}^{TN_3}} \tag{3}$$

Figures 8a–c illustrate the coverage areas of a network having 9  $TNs$  (including the sink node), each  $TN$  is responsible for managing a number of  $SNs$ . Figures 8a, 8b and 8c

show the resulting gray areas associated to each  $TN$  for a delta value of 1, 5 and 10 units, respectively. Again only nodes located in gray areas should disagree with the assertion while nodes located in non-gray areas remain silent. As we can observe in the figures, gray areas become larger as the delta value decreases in this example, which is expected since smaller delta values reduce the area in which  $SNs$  remain silent.

Now let  $A_{reply}^{TN_i}$  be the associated gray area observed by  $TN_i$ . Let  $m + 1$  be the number of  $TNs$  in the network, including  $TN_0$ , and let us recall that each  $SN$  is associated to a  $TN$ , then the percentage of  $SNs$  in the network that remain silent during a query can be approximated as

$$Percentage_{silent} \approx \frac{1}{m + 1} \sum_{i=0}^m \frac{A_{silent}^{TN_i}}{A_{coverage}^{TN_i}} \times 100 \quad (4)$$

In Fig. 9 we show this percentage for various values of delta. In this figure we consider a sensor network with

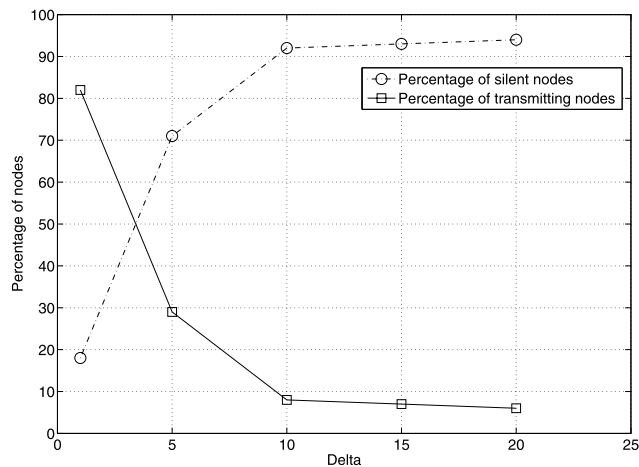


Fig. 9 Percentage of silent and transmitting nodes

50 nodes, transmission range was 10 meters and data readings among neighboring nodes were highly correlated. As we can observe, even for a delta value of 1, 18 % of the nodes remained silent in this example, while 92 % remained silent for a delta value of 10 units. If we now recall that in the traditional cluster head approach 100 % of the sensors are required to transmit, we can foresee the potential of Disagree in terms of energy savings.

### 5 Implementation, simulations, experiments and results

We implemented Disagree in NS2.29. For phase 1 we implemented the minimum tree utilized to propagate queries in the network layer according to Prim’s minimum spanning tree algorithm [10].

For the experiments we created various sensing profiles with different characteristics in order to study the performance of Disagree. We selected three different sensing profiles (named mosaic, Gaussian, and random), that exhibit different spatial data redundancy characteristics. We use a gray scale to graphically represent data readings from 0 to 100 units (see Fig. 10 top), and the sink node was always located in the center of the network.

*Mosaic* This sensing profile consists of a fixed grid with one hundred squares (10 m by 10 m) that covered the entire sensor network. A fixed interval of ten-unit values was assigned randomly to each square. For instance, if a square was given the interval 70–80 units, a sensor node located within this square obtained its reading by picking a value randomly within the 70–80 unit interval. See Fig. 10a.

*Gaussian* For creating this profile we used a Gaussian-like equation as shown by (5) and illustrated in Fig. 10b. A sensor node located at position  $(x, y)$  needs only to input its

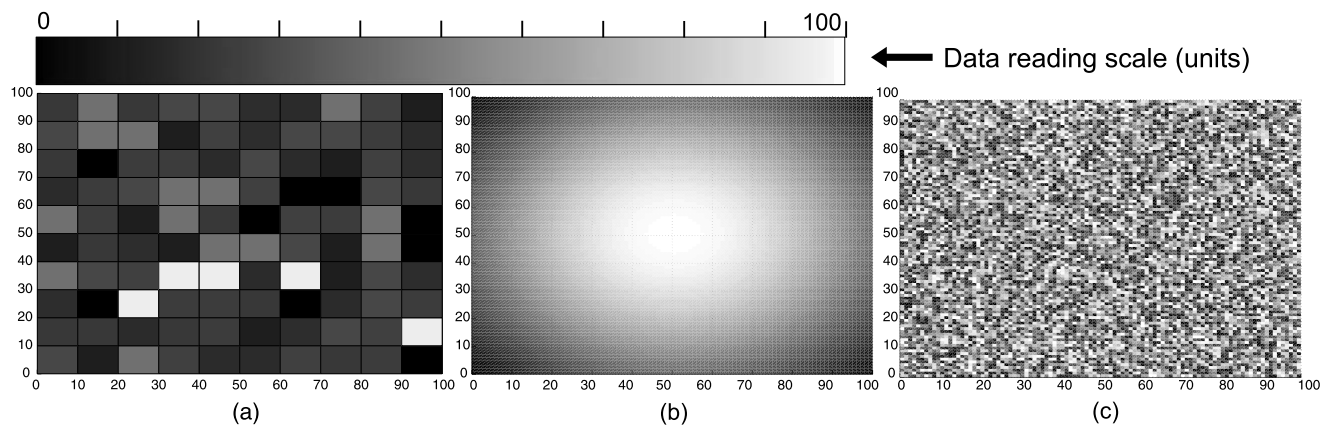


Fig. 10 Data readings scale. (a) Mosaic, (b) Gaussian and (c) Random field

**Table 1** Equivalences between  $\Delta$ (Disagree) and  $\tau$ (CAG)

Reading = 20		Reading = 50		Reading = 80	
$\Delta$ (Disagree)	$\tau$ (CAG)	$\Delta$ (Disagree)	$\tau$ (CAG)	$\Delta$ (Disagree)	$\tau$ (CAG)
0	0 %	0	0 %	0	0 %
1	10 %	1	1 %	1	1 %
10	50 %	10	20 %	10	20 %
20	–	20	40 %	20	30 %
100	–	100	–	100	–

coordinates in Eq. (5) to obtain its reading.

$$Reading(x, y) = 100e^{-(x-50)^2 - (y-50)^2} * .0002 \quad (5)$$

We use this function only as an example of a sensing profile where there are smooth variations of data readings among neighboring nodes.

*Random* In this sensing profile, each sensor node picks a value randomly between 0 and 100 units independently of other sensors, see Fig. 10c. In this scenario data readings among neighboring nodes are highly uncorrelated and variations of data readings among neighboring nodes can be large.

### 5.1 Computer simulations

For the simulations we created various topologies with different characteristics, such as node density, radio propagation range, data redundancy and delta values. We considered 50, 100, 150, 200, 300, 400 and 500 nodes, the radio range was 10 and 20 meters and we considered delta values of 0, 10, 20, and 100 units. We compared our results with LEACH [7] requiring a full view of the network where all *SNs* are required to transmit their readings to their associated *TN*. For LEACH we used the same tree used by Disagree. Additionally, we compared Disagree with CAG [38] using threshold values of 20 %, 10 % and 0 %. For the three protocols, each cluster head waits until it hears from each and every upstream cluster head before transmitting all collected data back to the sink node. Additionally, for each *event* the sink node requested a full view of the network and we rotated cluster heads every 10 events.

For each simulation we counted the number of transmitted packets, number of silent nodes, and estimation errors during all phases of the three protocols being compared. While the concept of delta in Disagree is similar to the concept of error threshold in CAG, they are not directly equivalent (Table 1 shows some equivalences between delta and error thresholds for various readings).

Figures 11a, 11b and 11c show the number of transmitted packets for Gaussian, random and mosaic scenarios considering all phases of the three protocols. Considering the

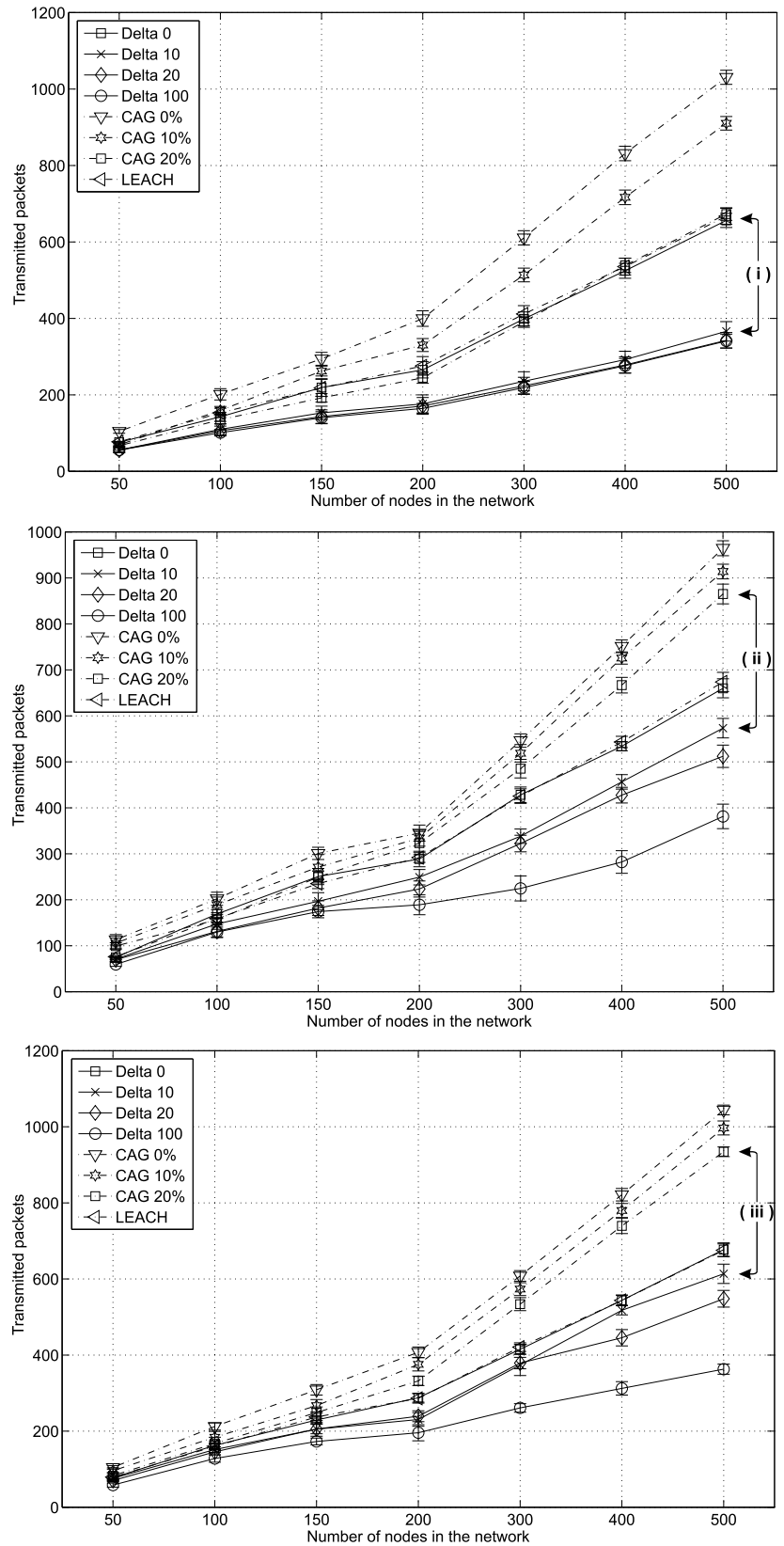
performance of Disagree, we can observe that the number of transmitted packets increases as the number of nodes also increases for any delta value. This happens because having more nodes in the network leads to a bigger, and potentially more disagreements from sensor nodes. In these figures, the largest number of transmitted packets occurs when delta = 0 units, which is the result of sensor nodes always transmitting their readings to their associated *TNs*. On the other hand, when the value of delta increases, the number of transmitted packets decreases as more sensors remain silent. The main difference among these three figures is that for the Gaussian scenario, curves for delta equal to 100, 20 and 10 units provided a similar number of transmitted packets, as opposed to the other two sensing fields where delta values of 20 and 10 units provided an increased number of transmitted packets. This situation happens because while most sensor nodes remained silent in the Gaussian scenario for delta = 10, few sensors disagreed in the mosaic scenario and almost all sensors disagreed in the random scenario for the same value of delta.

We can observe in these figures that CAG produces a behavior similar to Disagree. As we mentioned before, curves for Disagree and CAG are not directly comparable. However, in order to facilitate a comparison we indicate with an arrow (see arrows i, ii and iii in Fig. 11) two curves that we consider are somewhat equivalent (see Table 1). The results pointed out by the arrows indicate that CAG produces more signaling packets than Disagree for the three scenarios including the case when  $\Delta = 0$  and  $\tau = 0$ . This result is mainly due to the fact that in Disagree there is a larger number of silent nodes and there is a smaller distribution tree as compared to CAG.

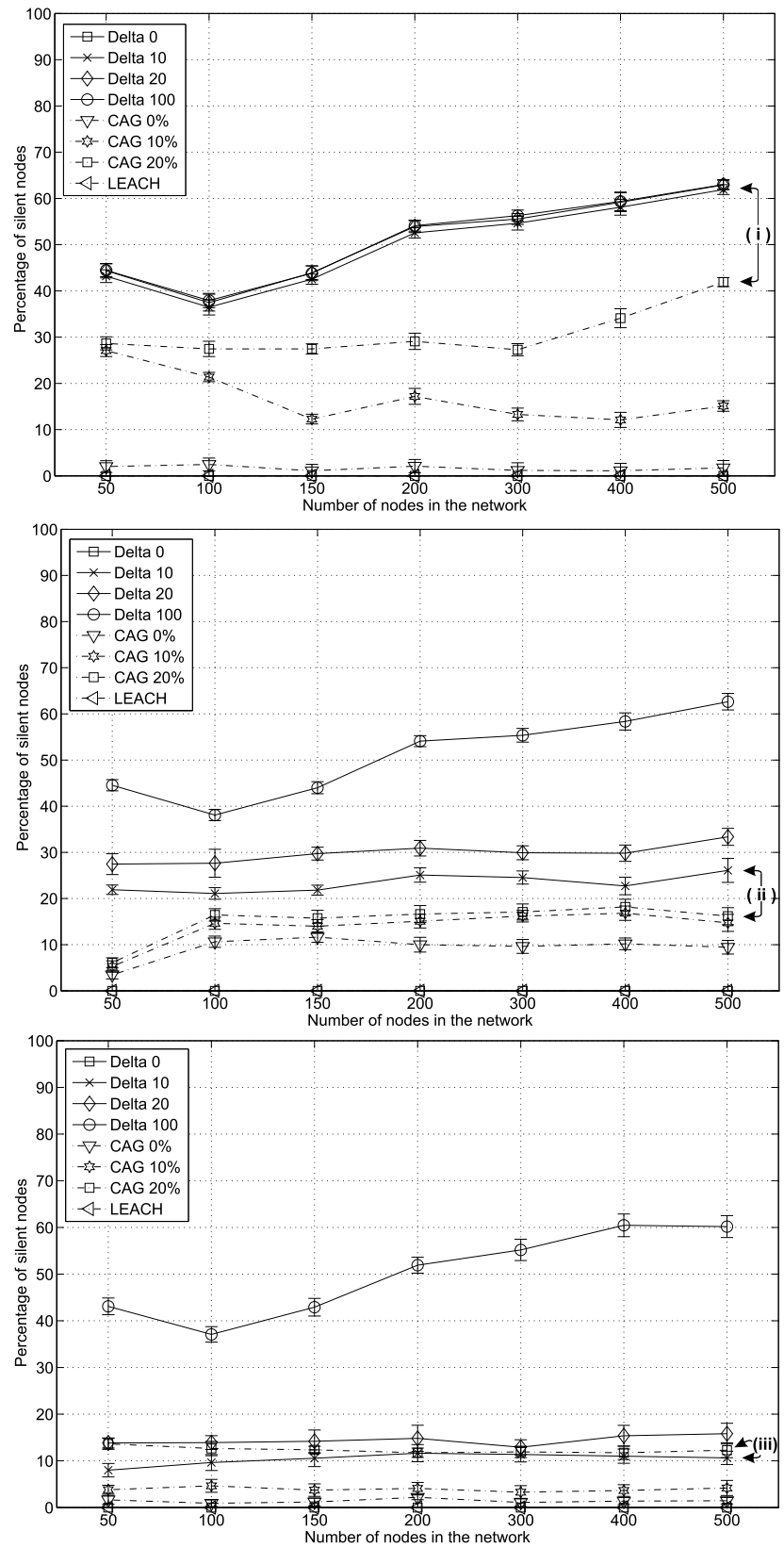
Finally, for the LEACH protocol the number of transmitted packets falls somewhere between DISAGREE and CAG curves. As we can see in these figures, Disagree outperforms LEACH except for delta = 0, where both protocols produced similar results. This is mainly the result of having silent nodes in Disagree, as opposed to LEACH where nodes always transmit. To achieve energy savings it is recommended to use Disagree with  $\Delta > 0$ . On the other hand, the LEACH approach outperforms CAG for any  $\tau$  value, and this happens because LEACH has a smaller distribution tree. It is important to mention that in the implementation of LEACH we used a minimum distribution tree. In case the distribution tree deviates from the minimum tree, LEACH will produce more signaling packets than the ones presented in these figures.

Figures 12a, 12b and 12c show the percentage of silent nodes for the three sensing fields being considered. As expected, more sensor nodes remained silent as the value of delta increases for Disagree, and this percentage does not change much even if the number of nodes in the network varies. While these figures exhibit similar trends, the most

**Fig. 11** Transmitted packets:  
**(a)** Gaussian field, **(b)** Mosaic field and **(c)** Random field with 95 % confidence intervals

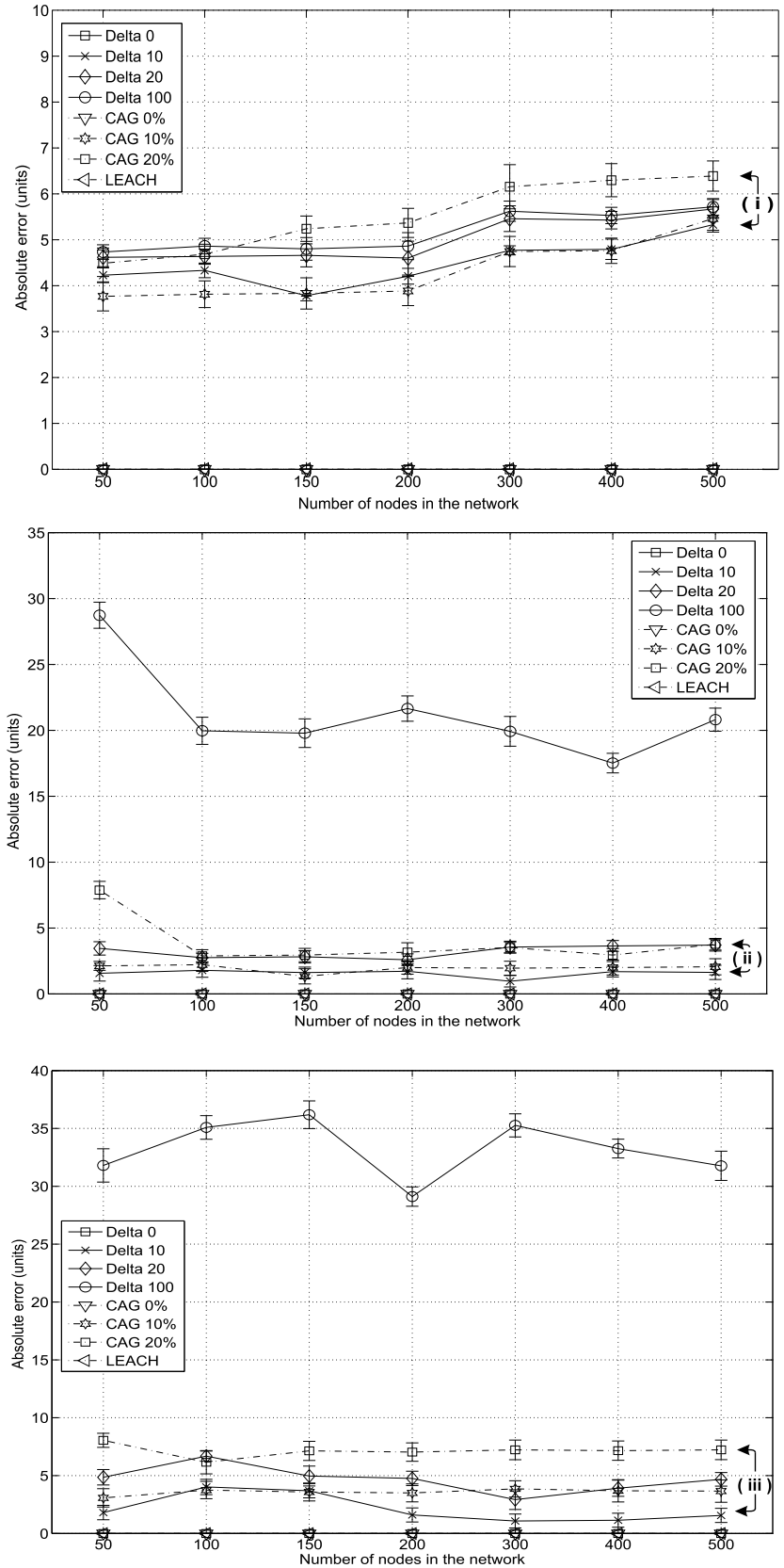


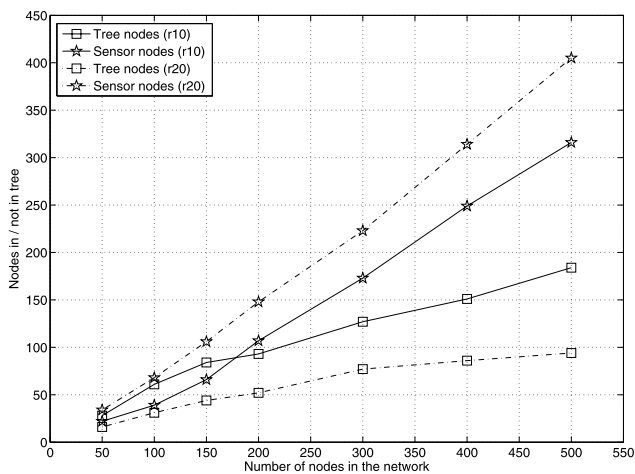
**Fig. 12** Percentage of silent nodes: (a) Gaussian field, (b) Mosaic field and (c) Random field with 95 % confidence intervals





**Fig. 13** Estimation error (units): **(a)** Gaussian field, **(b)** Mosaic field and **(c)** Random field with 95 % confidence intervals





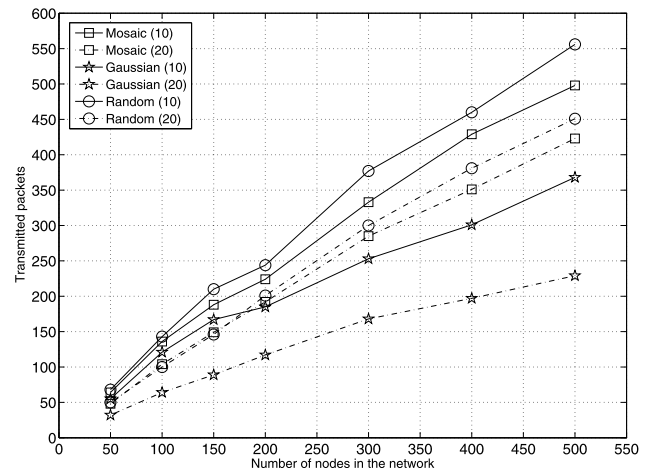
**Fig. 14** Number of tree nodes and sensor nodes

important difference happens again for the Gaussian scenario. As we can see in these figures about 50 % of the nodes remained silent in the Gaussian scenario for delta values of 10, 20 and 100 units, as opposed to the other two sensing fields where a delta value of 20 provided 30 % and 20 % of silent nodes for the mosaic and random scenarios, respectively. For CAG we observe that (see the data values pointed out by arrows i, ii and iii), there are fewer silent nodes compared with Disagree, which again is the result of CAG having a larger distribution tree. Finally, for LEACH we can observe that the percentage of silent nodes is always 0 % as all nodes are always required to transmit.

Figures 13a, 13b and 13c show the absolute error. This estimation error is the absolute difference between the estimated and the real readings at each sensor node. As we can observe, for delta values of up to 10 units, the absolute error is quite low for all three scenarios in Disagree (within 5 units), and only for a delta equal to 100, the error jumps above 20 units for the mosaic and random scenarios. The same behavior can be observed in CAG but again we observe that Disagree achieves lower estimation errors compared with CAG. In LEACH there is no estimation error because all nodes transmit their exact readings. Again the equivalent spatial resolution is indicated with arrows i, ii and iii.

Summarizing the observed behavior on all these figures we can see that on one side we have the Gaussian scenario for which there are few disagreements combined with small estimation errors, and on the other side we have the random scenario for which for any delta value there are lots of disagreements and higher estimation errors. More important, even for the random scenario, Disagree outperforms LEACH method that requires all sensor nodes to report their readings to their associated cluster head.

In Disagree the transmission range impacts the behavior of the protocol in two ways. Increasing the transmis-



**Fig. 15** Scenario comparison

sion range reduces the number of *TNs* in the tree but increases the number of associated *SNs* per *TN*. Similarly, a larger transmission range increases the probability that distant *SNs* disagree with an assertion packet as spatial correlation decreases with distance. In Fig. 14 we show how the transmission range impacts the number of transmitted packets for Disagree only; we consider values of 10 and 20 meters, respectively. In general for any scenario curves for  $R = 20$  were below the equivalent curve for  $R = 10$ . This is mainly the result of having a smaller tree for  $R = 20$  (see Fig. 15), and also because the number of sensor nodes reporting to a tree node increased, nodes that can potentially remain silent for larger delta values.

Finally, Fig. 16 shows the number of times, during a 10 epoch interval, each node was part of the tree in a network of 200 nodes. We can observe that the majority of nodes (78 %) were tree nodes 5 times. Some nodes (19 %) were never tree nodes because these nodes were located in the boundary of the network. Finally, because we did not rotate the sink node, this node was always part of the tree. As we can observe in Fig. 16 the tree node role is homogeneously distributed among nodes.

### 5.1.1 Energy consumption

We use the energy consumption model used in [7] for computing the energy consumption while transmitting and receiving  $l$  bits over  $d$  meters between transmitter and receiver. In this model if a sensor node transmits over a distance less than  $d_0$  the model uses the free space model  $\epsilon_{fs}$ , else it will use the multipath fading model  $\epsilon_{mp}$ . Thus, to transmit an  $l$ -bit message over a distance  $d$ , the radio spends

$$E_{Tx}(l, d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2, & d < d_0 \\ lE_{elec} + l\epsilon_{mp}d^4, & d \geq d_0 \end{cases} \quad (6)$$

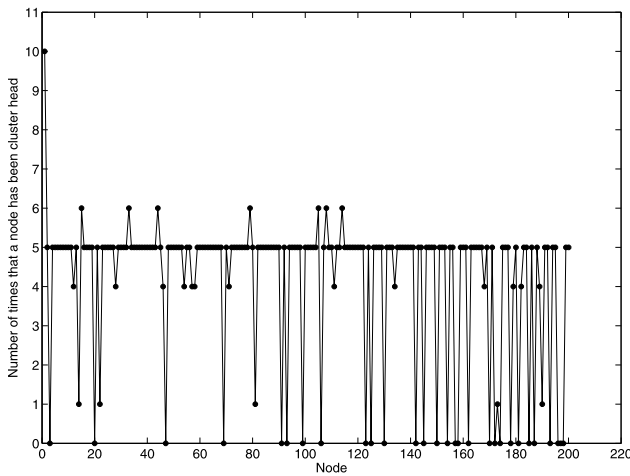


Fig. 16 Tree node role rotation

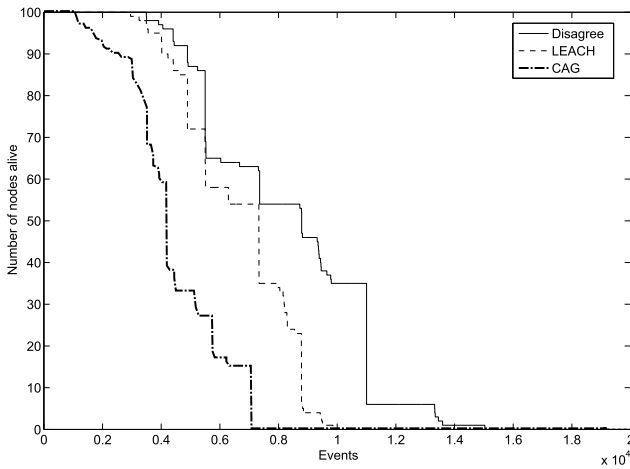


Fig. 17 Number of nodes alive

and to receive this message, the radio spends

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \tag{7}$$

For the simulation described in this paper, the communication energy parameters are set as:  $E_{elec} = 50$  nJ/bit,  $\epsilon_{fs} = 10$  pJ/bit/m<sup>2</sup> and  $\epsilon_{mp} = 0.0013$  pJ/bit/m<sup>4</sup>, similar to [7].

In order to compare Disagree with CAG and LEACH in terms of energy efficiency, we simulated the three protocols during 20000 events and rotated cluster heads every 10 events. Similar to [7] we run simulations until the last node died. The results are shown in Fig. 17 where we can see that network lifetime in Disagree is longer compared with either CAG or LEACH. CAG presents the shortest network lifetime because rotation of the cluster heads necessarily involves changing the topology, which requires all nodes to forward packets to choose new cluster, bridge and non-participating nodes.

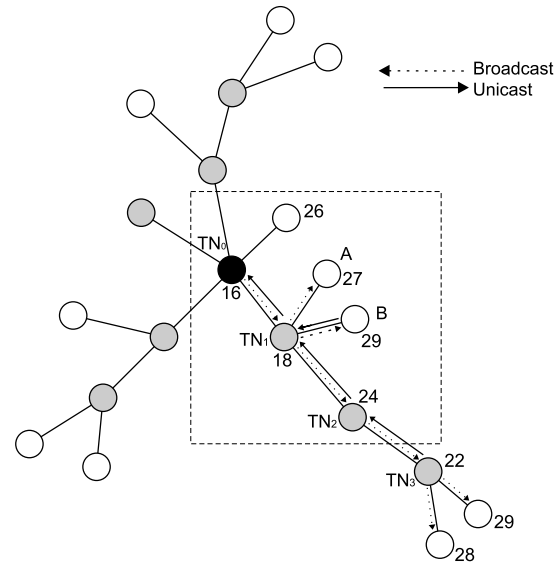


Fig. 18 Testbed experiment

Table 2 The lifetime of nodes

Node	Lifetime (h)
$TN_0$	140
$TN_1$	152
$TN_2$	115
$SN_A$	215
$SN_B$	147

### 5.2 Testbed experiments

We implemented Disagree in a platform composed of tmote sky and telosb nodes. Figure 18 illustrates the location of nodes, the resulting tree used during the expansion and contraction phases of Disagree, and also the readings at each node. Readings did not change during the experiments, and the value of delta was 10. The objective of these experiments was to compare energy consumption for tree and sensor nodes using Disagree. The sink node requested full views of the network every 10 seconds. We focus now on  $TN_1$  having two associated sensor nodes:  $SN_A$  never disagreed while  $SN_B$  always disagreed. Lifetime of nodes associated to  $TN_1$  is shown in Table 2. In this table we can observe that lifetime of  $SN_A$ , lasted 30 % more compared with  $SN_B$ . This result clearly shows the energy saving potential of Disagree. This table also shows the lifetime of tree nodes, emphasizing the need to rotate the tree role periodically to avoid tree nodes dying early.

### 6 Conclusions

In this paper we introduce Disagree, a new data acquisition technique for WSNs that exploits spatial data correlation at

the sensor level. Rather than asking all sensor nodes to reply to queries requesting a full view of the network, Disagree requires sensors to reply to the query only if their readings fall outside a smaller interval centered at the reading reported by their associated tree node as the query is propagated in the network. We show that this behavior is the core of an energy-efficient way to retrieve data in WSNs. Readings from silent sensor nodes can be easily estimated, and the estimation error can be controlled tuning a protocol parameter. We implemented Disagree in NS2.29 and performed various experiments to test its advantages and drawbacks under various sensing scenarios. Results indicate performance increases (i.e., fewer packets are transmitted) as data reading variations among neighboring nodes become smaller which is a common property of many physical phenomena. In the worst case, Disagree performs in the same way as a traditional cluster head method where all sensors reply to queries requesting a full view of the sensing network.

**Acknowledgements** This work was supported in part by research funds from UNAM/PAPIIT Grants' IN106609, IN114813 and CONACYT Grants 105117.

## References

- Akkaya, K., & Younis, M. (2005). A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3), 325–349.
- Akyildiz, I. F., Su, W., Sankarasubramanian, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *IEEE Communications Magazine*, 40(8), 102–114.
- Al-Karaki, J. N., & Kamal, A. E. (2004). Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6), 6–28.
- Barencio, C., Gonzalez, R., Cardenas, N., & Garcia, L. J. (2008). A proposal of a wireless sensor network routing protocol. *Telecommunication Systems*, 38(1), 61–68.
- Boyinbode, O., Le, H., Mbogho, A., Takizawa, M., & Poliah, R. (2010). A survey on clustering algorithms for wireless sensor networks. In *Proceedings of 13th international conference on network-based information systems*, Takayama, Japan (pp. 358–364).
- Braginsky, D., & Estrin, D. (2002). Rumor routing algorithm for sensor networks. In *Proceedings of the 1st ACM international workshop on wireless sensor networks and applications*, Shenzhen, China (pp. 22–31).
- Chandrakasan, A., Smith, A., & Heinzelman, W. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4), 660–670.
- Chen, J., Guan, Y., & Pooch, U. (2005). A spatial-based multi-resolution data dissemination scheme for wireless sensor networks. In *Proceedings of the 19th IEEE international parallel and distributed processing symposium*, Denver, USA (pp. 245–253).
- Chu, D., Deshpande, A., Hellerstein, J., & Hong, W. (2006). Approximate data collection in sensor networks using probabilistic models. In *Proceedings of the 22nd international conference on data engineering*, Atlanta, USA (pp. 48).
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms*. McGrawHill: MIT Press.
- Dai, F., & Wu, J. (2006). On constructing k-connected k-dominating set in wireless ad hoc and sensor networks. *Journal of Parallel and Distributed Computing*, 66(7), 947–958.
- Du, D. Z., & Pardalo, P. (2005). Connected dominating set in sensor networks and MANETs. In *Handbook of combinatorial optimization* (pp. 329–370). Berlin: Springer.
- Fasolo, E., Rossi, M., Widmer, J., & Zorzi, M. (2007). In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, 14(2), 70–87.
- Ganesan, D., Estrin, D., & Heidemann, J. (2003). DIMENSIONS: why do we need a new data handling architecture for sensor networks? *ACM SIGCOMM Computer Communications*, 33(1), 143–148.
- Guan, X., Guan, L., Wang, X., & Ohtsuki, T. (2010). A new load balancing and data collection algorithm for energy saving in wireless sensor networks. *Telecommunication Systems*, 45(4), 313–322.
- Gupta, H., Navda, V., Das, S., & Chowdhary, V. (2008). Efficient gathering of correlated data in sensor networks. *ACM Transactions on Sensor Networks*, 4(1), 1–31.
- Haas, Z. J., Halpern, J. Y., & Li, L. (2006). Gossip-based ad hoc routing. *IEEE/ACM Transactions on Networking*, 14(3), 479–491.
- Iima, Y., Kanzaki, A., Hara, T., & Nishio, S. (2009). Overhearing based data transmission reduction for periodical data gathering in wireless sensor networks. In *Proceedings of the international conference on complex, intelligent and software intensive systems*, Fukuoka, Japan (pp. 1048–1053).
- Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., & Silva, F. (2003). Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking*, 11(1), 2–16.
- Israr, N., & Awan, I. (2008). Coverage based inter cluster communication for load balancing in heterogeneous wireless sensor networks. *Telecommunication Systems*, 38(3), 121–132.
- Jin, Y., Chen, F., Che, G., & Hu, W. (2010). Energy-efficient data collection protocol for wireless sensor network based on tree. In *Proceedings of the Asia-Pacific conference on wearable computing systems*, Shenzhen, China (pp. 82–85).
- Kondo, S., Kanzaki, A., Hara, T., & Nishio, S. (2011). Energy-efficient data gathering using sleep scheduling and spatial correlation based on data distribution in wireless sensor networks. In *Proceedings of the 14th international conference on network-based information systems*, Tirana, Albania (pp. 194–201).
- Kour, H., & Sharma, A. K. (2010). Hybrid energy efficient distributed protocol for heterogeneous wireless sensor network. *International Journal of Computer Applications*, 4(5), 37–41.
- Lee, K., Lee, J., Lee, H., & Shin, Y. (2010). A density and distance based cluster head selection algorithm in sensor networks. In *Proceedings of the 12th international conference on advanced communication technology*, Phoenix, USA (pp. 162–165).
- Lee, S., Lee, C., Cho, Y., & Kim, S. (2004). A new data aggregation algorithm for clustering distributed nodes in sensor networks. *Lecture Notes in Computer Science*, 3262, 508–520.
- Madden, S., Franklin, M., Hellerstein, J., & Hong, W. (2002). TAG: a tiny aggregation service for ad-hoc sensor networks. In *Proceedings of the 5th symposium on operating systems design and implementation*, Boston, USA (pp. 131–146).
- Madden, S., Franklin, M., Hellerstein, J., & Hong, W. (2005). TinyDB: an acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1), 122–173.
- Manjeshwar, A., & Agrawal, D. P. (2001). TEEN: a protocol for enhanced efficiency in wireless sensor networks. In *Proceedings of the 15th international parallel and distributed processing symposium*, San Francisco, USA (pp. 2009–2015).
- Manjeshwar, A., & Agrawal, D. P. (2002). APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval



- in wireless sensor networks. In *Proceedings of the 16th international parallel and distributed processing symposium*, Florida, USA (pp. 195–202).
30. Ni, S., Tseng, Y., Chen, Y., & Sheu, J. P. (1999). The broadcast storm problem in a mobile ad-hoc network. In *Proceedings of the 5th annual ACM/IEEE international conference on mobile computing and networking*, Seattle, USA (pp. 151–162).
  31. Petrovic, D., Shah, R., Ramchandran, K., & Rabaey, J. (2003). Data funneling: routing with aggregation and compression for wireless sensor networks. In *Proceedings of the 1st IEEE international workshop on sensor network protocols and applications*, Anchorage, USA (pp. 156–162).
  32. Rajagopalan, R., & Varshney, P. K. (2006). Data aggregation techniques in sensor networks: a survey. *IEEE Communications Surveys and Tutorials*, 8(4), 48–63.
  33. Roxin, A., Gaber, J., Wack, M., & Nait-Sidi-Moh, A. (2007). Survey of wireless geolocation techniques. In *Proceedings of the global communications conference*, Washington, USA (pp. 1–9).
  34. Vuran, M. C., Akan, C. B., & Akyildiz, I. F. (2004). Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 45(3), 245–259.
  35. Vuran, M. C., & Akyildiz, I. F. (2006). Spatio correlation-based collaborative medium access control in wireless sensor networks. *IEEE Transactions on Networking*, 14(2), 316–329.
  36. Wang, Y., Hsieh, Y., & Tseng, Y. (2009). Multiresolution spatial and temporal coding in a wireless sensor networks for long-term monitoring applications. *IEEE Transactions on Computers*, 58(6), 827–838.
  37. Yao, Y., & Gehrke, J. (2002). The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Record*, 31(3), 9–18.
  38. Yoon, S., & Shahabi, C. (2007). The clustered aggregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks. *ACM Transactions on Sensor Networks*, 3(1), 1–39.
  39. Younis, O., & Fahmy, S. (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4), 366–379.



**Martha Montes-de-Oca** received her B.Sc. with honors in Informatics and the M.Sc. in Computer Engineering from the National Autonomous University of Mexico (UNAM). She has worked as Server and Network Manager and Programmer Senior. She is currently a Ph.D. candidate at UNAM. Her research interests cover wireless sensor networks and energy efficiency in data collection.



**Javier Gomez** received the BS degree with honors in Electrical Engineering in 1993 from the National Autonomous University of Mexico (UNAM) and the M.S. and Ph.D. degrees in Electrical Engineering in 1996 and 2002, respectively, from Columbia University and its COMET Group. During his Ph.D. studies at Columbia University, he collaborated and worked on several occasions at the IBM T.J. Watson Research Center, Hawthorne, New York. His research interests cover routing, QoS, and MAC design for wireless ad hoc, sensor, and mesh networks. Since 2002, he has been an Assistant Professor with the National Autonomous University of Mexico. Javier Gomez is member of the SNI (level I) since 2004.



**Miguel Lopez-Guerrero** received his B.Sc. in Mechanical-Electrical Engineering in 1995 and the M.Sc. in Electrical Engineering in 1998, both from the National Autonomous University of Mexico. He received his Ph.D. in Electrical Engineering from the University of Ottawa in 2004. He is an Associate Professor with the Metropolitan Autonomous University (Mexico City). His research interests span several aspects of telecommunication networks including network traffic modeling, medium access control and mobility-related studies. He is a member of the IEEE.