Computer Networks 53 (2009) 1980–1996

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A reactive role assignment for data routing in event-based wireless sensor networks ☆

Eduardo F. Nakamura [a,*], Heitor S. Ramos [b], Leandro A. Villas [b], Horacio A.B.F. de Oliveira [c], Andre L.L. de Aquino [b], Antonio A.F. Loureiro [b]

[a] Computer Science Lab, Research and Technological Innovation Center (FUCAPI), Danilo Areosa de Matos, 381, 69075-351, Manaus, AM, Brazil
[b] Department of Computer Science, Federal University of Minas Gerais, PO Box 702, 30123-970, Belo Horizonte, MG, Brazil
[c] Department of Computer Science, Federal University of Amazonas, Rodrigo Octavio Jordao, 3000, Setor Norte, 69077-000, Manaus, AM, Brazil

ARTICLE INFO

Article history:
Received 16 January 2008
Received in revised form 9 February 2009
Accepted 5 March 2009
Available online 28 March 2009

Responsible Editor: M.C. Vuran

Keywords:
Information fusion
Role assignment
Wireless sensor networks
Data routing
In-network aggregation

ABSTRACT

In this work, we show how we can design a routing protocol for wireless sensor networks (WSNs) to support an information-fusion application. Regarding the application, we consider that WSNs apply information fusion techniques to detect events in the sensor field. Particularly, in event-driven scenarios there might be long intervals of inactivity. However, at a given instant, multiple sensor nodes might detect one or more events, resulting in high traffic. To save energy, the network should be able to remain in a latent state until an event occurs, then the network should organize itself to properly detect and notify the event. Based on the premise that we have an information-fusion application for event detection, we propose a role assignment algorithm, called Information-Fusion-based Role Assignment (InFRA), to organize the network by assigning roles to nodes only when events are detected. The InFRA algorithm is a distributed heuristic to the minimal Steiner tree, and it is suitable for networks with severe resource constraints, such as WSNs. Theoretical analysis shows that, in some cases, our algorithm has a $O(1)$-approximation ratio. Simulation results show that the InFRA algorithm can use only 70% of the communication resources spent by a reactive version of the Centered-at-Nearest-Source algorithm.

1. Introduction

Wireless sensor networks (WSNs) are strongly limited regarding power resources and computational capacity [2]. In addition, these networks need to autonomously adapt themselves to eventual changes resulting from external interventions, such as topological changes, reaction to a detected event, or requests performed by an external entity. Commonly, sensor networks are deployed to monitor the occurrence of specific events, such as fire, air condition, or presence of military targets. An important task in event-driven sensor networks is to efficiently deliver event data to the sink. Consequently, data routing is a fundamental task, which is commonly performed in a multihop fashion due to radio-range limitations and energy constraints. Usually, roles are assigned to sensor nodes so each node knows how to participate in the execution of a given task.

The role assignment problem occurs in team-based applications where the involved entities take different roles that demand different resources to accomplish different tasks. The main challenge is how to reactively

☆ This work extends the original contribution proposed by Nakamura et al. [1]. The current version includes several new contributions: a more complete description of the proposed algorithm; additional comparisons with other algorithms; and a theoretical analysis that yields in defining performance bounds for the proposed algorithm.

* Corresponding author. Tel.: +55 92 2127 3052; fax: +55 92 3613 2655.

E-mail addresses: eduardo.nakamura@fucapi.br, efnaka@gmail.com (E.F. Nakamura), hramos@dcc.ufmg.br (H.S. Ramos), leandro@dcc.ufmg.br (L.A. Villas), horacio@ufam.edu.br, horacio@site.uottawa.ca (H.A.B.F. de Oliveira), alla@dcc.ufmg.br, alla@iceb.ufop.br (A.L.L. de Aquino), loureiro@dcc.ufmg.br (A.A.F. Loureiro).

URL: http://www.nakamura.eti.br/eduardo (E.F. Nakamura).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.03.009

change roles in response to the dynamic situations that are identified. In the context of WSNs, a role assignment may be triggered by different reasons such as event detections, failure occurrences, and management tasks. Also, role assignment may be performed with different objectives such as density control, energy balancing, and information fusion.

In this work, we use role assignment to find a minimum transmission tree that maximizes in-network data aggregation. Current solutions for this problem [3–6] try to optimize the data-gathering task by proactively assigning roles independent of event occurrences, which leads to energy waste during periods of inactivity.

Thus, a major contribution of this work is an event-based role assignment algorithm that tries to reactively find the shortest routes (connecting source nodes to the sink) that maximize data aggregation. This algorithm, called Information-Fusion-based Role Assignment (InFRA), establishes a hybrid network organization in which source nodes are organized into clusters and the cluster-to-sink communication occurs in a multihop fashion. The resulting topology is a distributed heuristic to the minimal Steiner tree. Although there are other heuristics for the Steiner tree for routing in WSN [7–10], the InFRA algorithm is a distributed heuristic that aims at providing a reasonable approximation ratio with acceptable communication cost. However, as we show further on in Section 7.8, if the event duration is not long enough, a simple shortest-path tree might be a better choice.

Theoretical analysis shows that our algorithm has a $O(1)$-approximation ratio when the network diameter remains constant and, in large-scale networks it has a k-approximation ratio, where k is the number of simultaneous events. Simulation results show that InFRA can save communication resources compared to a proactive and other reactive algorithms. In some cases, the InFRA algorithm uses only 70% of the communication resources spent by a reactive version of the Centered-at-Nearest-Source algorithm [10].

For the sake of simplification, we consider a lossless channel model in our theoretical analysis and simulations. Such an assumption allows us to find the theoretical bounds that define some performance limits. To be fair, all algorithms were analyzed and evaluated under the same premise. A lossy channel will affect every aggregation solution. In general, algorithms that aggregate more data flows are more affected by lossy channels. When we have lossy channels, we should adopt some mechanism to overcome packet losses, such as retransmissions or data prediction. The application requirements will determine whether or not it is worth implementing such mechanisms.

The remainder of the paper is organized as follows. Section 2 discusses some role assignment solutions. Section 3 presents the background knowledge supporting this work. The problem of finding the minimum transmission routing tree is formalized in Section 4. Our role assignment algorithm (InFRA) is presented in Section 5. Theoretical results of our heuristic are presented in Section 6, and simulation results in Section 7. In Section 8, we present our final remarks.

2. Related work

Bonfils and Bonnet [4] propose an adaptive and decentralized solution that progressively refines the role assignment by evaluating neighbor nodes. Their solution looks for the role assignment that minimizes the amount of data transmitted in the network. At regular intervals, a cost function is evaluated and the role migrates to the node with the lowest cost. The communication cost introduced by the solution is not considered by the authors, which is the key point of our solution in which we try to balance the communication cost with the quality of the routing tree.

The Sensor Placement and Role Assignment for Energy-Efficient Information Gathering (SPRING) algorithm [11] was proposed for mobile sensor networks. The problem that SPRING tries to solve relies on placing nodes and assigning roles to them so the system lifetime is maximized, ensuring that the region of interest is covered by at least one node. The SPRING algorithm outperforms a random placement, which is the minimum we could expect from a placement strategy. The main limitation of this solution is that it demands mobile sensor nodes, which can be too expensive and present mobility limitations depending on the terrain topology. InFRA is designed to regular WSNs in which we do not have mobile nodes.

Kochhal et al. [12] propose a role-based clustering algorithm that organizes the network by recursively finding connected dominating sets. These sets are used to define coordinators (cluster-heads) and routing nodes, and the remainder become sensing collaborators (sources). The clustering process considers the sensing ability of the nodes, so the detection capability of the clusters is enhanced.

The DFuse framework proposed by Kumar et al. [5] addresses the role assignment problem providing two modules: a fusion and a placement module. The fusion module allows the application to be built using a dataflow graph that specifies the roles of each node. The placement module maps this graph onto the network and migrates the roles according to a specified cost function. Frank and Römer [13] propose a generic role assignment framework that allows the user to specify roles and assignment rules. The framework defines three core elements. The first is a property directory used to access capabilities and parameters of the nodes. The second is the role specification. The third is the algorithm that assigns the roles based on the role specifications and properties of the nodes. From the scientific perspective, these role assignment frameworks could be used to implement, theoretically, any other role assignment algorithm, such as the solution we propose here.

Cristescu et al. [14] consider the problem of correlated data gathering by a network in which the goal is to minimize the total transmission in a routing tree. The key point of this work is that authors investigate two specific coding strategies: a Slepian–Wolf model and a joint entropy coding model with explicit communication. For both coding models the authors provide distributed algorithms. However, the Slepian–Wolf algorithms are complex and demand global network knowledge, usually prohibitive for WSNs. For the explicit communication case, they evaluate some heuristics, including the GIT [15], Leaves deletion, and a balanced SPT/TSP (shortest-path tree/travel-salesman paths). The criticism for these heuristics is that they demand the leaf nodes to know the shortest paths to every other sensor node. As we show in Section 6.2, although the algorithm cost is fair, the cost to get the necessary information is usually prohibitive in WSNs.

Ciciriello et al. [16] propose a scheme for routing data efficiently from multiple sources to multiple sinks, which is a slightly different problem than that addressed herein. The proposed decentralized scheme is based on a periodic adaptation of the message routes. To guarantee a high degree of overlapping among source–sink paths, each node can decide to locally manipulate a source–sink path by changing its parent. However, the parent switching is managed by using timeouts which introduce additional delays, not evaluated by authors.

Fan et al. [17] propose a semi-structured approach that uses a structureless technique locally followed by Dynamic Forwarding. In the proposed algorithm, the chosen structure, ToD (Tree on Directed acyclic graph), is composed of multiple shortest-path trees, and after performing local aggregation, nodes dynamically decide the forwarding tree based on the location of the source nodes. This defines a different type of heuristics that is based on the geographical location of sensor nodes. However, as Oliveira et al. [18] show, the errors of current localization algorithms are not irrelevant and can significantly affect routing solutions such as the one proposed by Fan et al. [17]. The solution proposed in this work, the InFRA algorithm, does not depend on location information. In general, algorithms based on the error-free knowledge of nodes' location are more efficient. However, they should be evaluated considering the errors of current localization algorithms.

3. Background

This section presents the background knowledge and concepts used in this work.

3.1. Network and event model

We consider a sensor network composed of n nodes of which one of them is the sink node. For the sake of simplification, we consider symmetric links, i.e., for any two nodes u and v, u reaches v if, and only if, v reaches u.

All events are static and described by an influence region (area). We assume a binary detection model, i.e., every node within the influence region of an event detects that event. Thus, we represent the network by the graph $G = (V, E)$ with the following properties:

- $V = \{v_1, v_2, \ldots, v_n\}$ is the set of sensor nodes, such that $|V| = n$ and v_1 is the sink node;
- $\mathbf{S} = \{s_1, s_2, \ldots, s_m\}$ is the set of sources, i.e., nodes detecting an event, such that $|\mathbf{S}| = m$ and $\mathbf{S} \subseteq V$;
- $\langle i,j \rangle \in E$ iff v_i and v_j are neighbors.

The **closed neighborhood** $\mathcal{N}$ is composed of the node itself and its neighbors. Thus, the closed neighborhood of node v_i is given by

$$\mathcal{N}_i = \{v_i\} \bigcup \left(\bigcup_{\langle i,j \rangle \in E} \{v_j\} \right). \tag{1}$$

In a sensor network, the network state is often used to guide decision-making processes. Alternatively, in localized algorithms, nodes make decisions based on the state of the node itself and the state of its neighbors.

Let $\mathbf{x}_i$ be the state of node $v_i \in V$. The network can be described by its state vector (**network state**), which can be defined as

$$\mathbf{X} = \bigcup_{v_i \in \mathbf{V}} \mathbf{x}_i. \tag{2}$$

The definition of the node state depends on the application. It can be a large set that includes all sorts of information about the node, such as residual energy, workload, bandwidth, noise level, and location; or a simple value, such as a flag indicating whether or not the node is a source. From the network state we can derive the **neighborhood state** of each node, which is a subset of the network state.

For each node $v_i \in \mathbf{V}$, we define its neighborhood state as

$$\mathbf{X}_i = \bigcup_{v_j \in \mathcal{N}_i} \mathbf{x}_j. \tag{3}$$

3.2. Deployment model

We consider that the node deployment results in a disturbed grid where the location of each node is disturbed by a random zero-mean Gaussian error. Therefore, nodes will tend to uniformly occupy the sensor field without forming a regular grid.

3.3. Role assignment model

In this section, we formalize the concepts of role and role assignment in the context of this work.

A **role** specifies the actions and computations executed by a node in the presence of a specific data stream and an identified condition. Thus, for a given node a role defines the expected behavior patterns associated with a particular data stream. A node may aggregate multiple roles only to process different data streams, i.e., a single node cannot have two different roles to process the same data stream.

As an example, we can have a network in which a node A can use a fusion role to process the data streams from nodes B and C, and use a relay role to process the data stream originated by node D. Alternatively, a node A can use a fusion role to process any temperature data, and a relay role to process humidity data.

The **space of roles**, or script, defines the set of all possible roles that can be assigned to a node and is represented by Ψ.

Let Δ be the set of all data streams produced by these sensor nodes.

Definition 3.1 (*Global Role Assignment – GRA*). In a GRA, roles are assigned based on the entire network state. Formally, a GRA is a surjective function $g : \mathbf{X} \times V \times \Delta \rightarrow \Psi$

that maps the network state, a node, and a data stream onto a role in the script.

A global role assignment demands the knowledge of the entire network state, which is often unfeasible. Typically, sensor nodes must make decisions based only on local information (node state) and local interactions (neighborhood state). Thus, a local role assignment is usually preferable to a global one.

Definition 3.2 (*Local Role Assignment – LRA*). In a LRA, roles are assigned based on the neighborhood state. Formally, a LRA is a surjective function $l : \mathbf{X}_i \times V \times \Delta \rightarrow \Psi$ that maps a neighborhood state, a node, and a data stream onto a role in the script.

The InFRA algorithm, herein proposed, is a LRA to find a minimum transmission tree. Thus, the idea is to have a LRA that assigns a role to a every node based on their neighborhood state. In the end, this role assignment defines a routing tree that tries to minimize the overall message exchange to notify events detected by the WSN. This problem and our proposed solution are better described in the following sections.

4. Problem statement

Let us assume that all nodes do not necessarily reach the sink node in one hop. Once an event is detected, we want to find a multihop routing structure that maximizes data aggregation with the minimum number of hops, i.e., a minimum transmission tree.

Definition 4.1 (*Problem definition*). Given a multihop network $G = (\mathbf{V}, \mathbf{E})$, we want to reactively find the minimum transmission tree connecting all $u \in \mathbf{S}$ to the sink node.

The minimum transmission tree is actually a minimum Steiner tree connecting the nodes that detect the event to the sink node, i.e., this is a NP-complete problem [19]. In this work, we provide a reactive solution that dynamically chooses the next hop minimizing the impact of eventual link losses. Our solution relies on reactively assigning roles when an event is detected.

5. InFRA: Information-Fusion-based Role Assignment

In our role assignment algorithm, when multiple nodes detect the same event, they organize themselves into clusters. Then, cluster-heads aggregate data from all cluster-members and send event data towards the sink. Since all nodes may not directly reach the sink node, notification packets are relayed in a multihop fashion. Our algorithm considers the following roles to set up a routing infrastructure:

- *sink* – node interested in a set of events;
- *collaborator* – node that detects an event (cluster-member);
- *coordinator* – node that detects an event and is responsible for notifying its occurrence (cluster-head);
- *relay* – node that forwards a data stream received by another node; every *relay* node also has an implicit *sensing* role, so it is still able to detect events.

Thus, our space of roles is $\Psi = \{sink, collaborator, coordinator, relay\}$.

When no event is being detected, all sensor nodes except the sink have the *relay* role. When at least one node detects an event, the role assignment algorithm executes the following procedures:

- Clusters are formed by assigning the *collaborator* and *coordinator* roles to nodes detecting events.
- The *relay* is assigned to the other nodes and routes are formed connecting clusters to the sink.
- Information is fused to reduce communication costs.

5.1. Cluster formation

The idea is to build clusters in such a way that we have only one cluster for each event being detected and cluster members are the detecting nodes. We might have different strategies to select the node with the *coordinator* role (cluster-head). For instance, we can choose the node with the smallest *id*, greatest degree, largest residual energy, shortest distance to the sink, or other metrics such as those suggested by Kochhal et al. [6]. Such information, used in the election process, is included in the node state that composes the network and neighborhood states defined in Section 3.1. For the sake of simplicity we choose the node with the smallest *id* because this strategy leads to smaller communication cost during the election phase. Thus, our node state will be $\mathbf{x}_i = \{id(v_i)\}$ for every $v_i \in \mathbf{V}$. The cluster formation algorithm is presented in Fig. 2.

This phase includes only the nodes that detect an event, i.e., the sources $u \in \mathbf{S}$. First, the nodes announce the event detection (line 3). This announcement does not assume any fixed MAC contention strategy; we schedule these nodes by introducing random delays to reduce the collision probability. Second, nodes assess their neighborhood, and the one with the smallest *id* becomes a *coordinator* (lines 4–8). Current coordinators announce their condition in an event-scoped flooding[1] (line 11). Then, only the *coordinator* of the smallest *id* keeps its role and floods its condition to the entire network, the other ones become *collaborators* (lines 14–18).

Fig. 1 depicts the clustering process. Fig. 1a depicts the communication range of the detecting nodes, and Fig. 1b the corresponding connectivity graph. Based on the neighborhood state, the nodes with the smallest *id* in their neighborhood become *coordinators* (nodes 1, 2, and 4 in Fig. 1c). However, only the *coordinator* of the smallest *id* keeps the role (node 1 in Fig. 1d).

The InFRA algorithm does not assume any fixed MAC contention strategy. In a previous evaluation [1] the algorithm was simulated considering a CSMA/CA MAC, in which we introduced small random delays to reduce the collision probability. Results show that this simple measure is statistically effective for overcoming such a problem, i.e.,

[1] An event-scoped flooding is a controlled flooding. It is essentially a traditional flooding where every node forwards a packet only once, but the nodes participating in the flooding are only the ones that have detected the event.

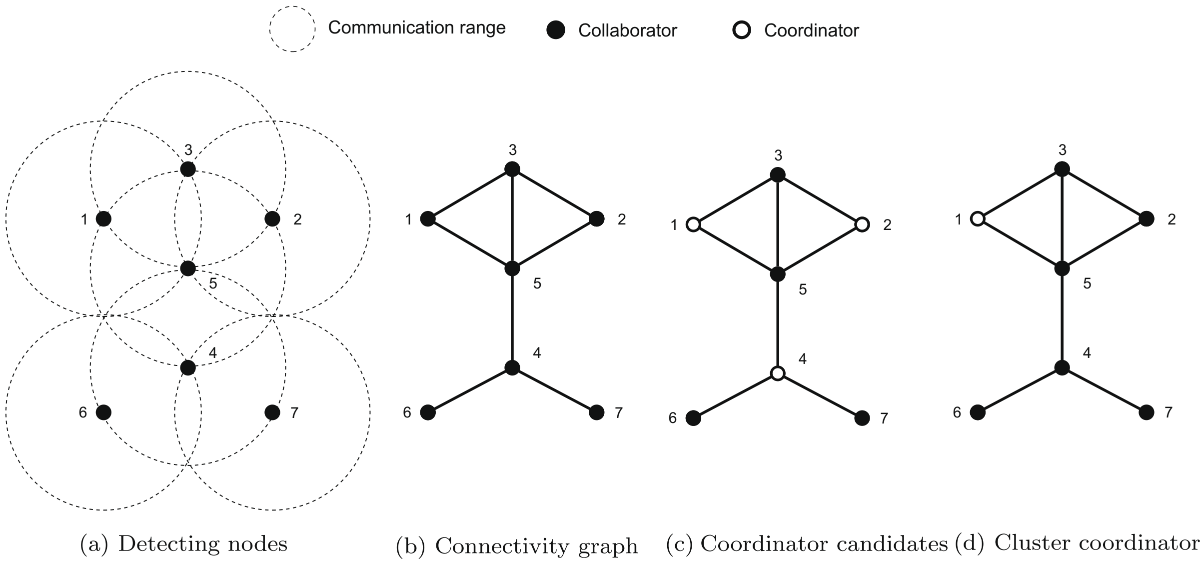


Fig. 1. Example of the clustering process.

although we eventually loose some announcements, which lead to sub-optimal solutions, such losses do not have a great impact on our solution, since it is already an approximated algorithm.

5.2. Route formation

Routes are formed by choosing the best neighbor at each hop. The function that defines the best neighbor depends on the application. In this case, we consider the best node as the one that leads to the shortest path to the sink and fuses as many clusters as possible, i.e., the resulting routes form a tree with the minimum number of edges connecting the *coordinators* to the sink node (see Section 4). This is done by choosing the neighbor closest to the sink node, and, in case of a tie, we choose the one that minimizes the distance to the other *coordinators*. This takes into account the aggregated coordinators-distance, which is defined as follows.

Definition 5.1 (*Aggregated Coordinators-Distance*). For every node $v_i \in \mathbf{V}$, its aggregated coordinators-distance, $dist-co(v_i)$, is the sum of the distances (in hops) between v_i and all *coordinator* nodes, i.e.,

$$dist-co(v_i) = \sum_{u \in CoordSet} distance(v_i, u), \qquad (4)$$

where $distance(v_i, u)$ is the distance in hops between nodes v_i and u, and *CoordSet* is the set of all *coordinator* nodes.

To understand how the aggregated coordinators-distance is computed, let us take a look at Fig. 4b. Once the clusters are formed, each coordinator floods a control message, and during this flooding, every node computes its distance to that coordinator. So, in Fig. 4b, nodes H, X, and O perform such flooding, randomly scheduled to reduce collisions. After this process, every node can compute the aggregated coordinators-distance. Thus, taking node B as an example, after the flooding processes, node B knows it is one hop from coordinator H, 4 hops from coordinator X, and three hops from coordinator O. Thus, the sum of all distances is eight hops (aggregated coordinators-distance).

The routing strategy is depicted in Fig. 3. First, the *relay* is assigned to the nodes that are neither *coordinators* nor *collaborators* (line 2). Then, the node chooses as the next hop a neighbor node closer to the sink and to the current *coordinators* (lines 4–11). When the node is ready it sends the aggregated data to the next hop (lines 12–14).

This routing strategy looks for the shortest path in such a way that the nodes in the relay process minimize the distance to the current *coordinators*. As a result, chances of route overlapping and, consequently, data aggregation are enhanced.

To illustrate the benefits of this routing strategy, let us consider the example of Fig. 4. When a simple shortest path is used, data aggregation may not occur because the shortest paths chosen for each cluster may not overlap (Fig. 4a). However, the routing strategy adopted by InFRA looks for the shortest path that leads to a minimum transmission tree, increasing the chances of data aggregation (Fig. 4b). In Fig. 4b, by using the InFRA algorithm, node L aggregates data streams from nodes H and X, and node F aggregates data streams from nodes L and O.

5.3. Information fusion

In the proposed network organization, we might have two different types of information fusion: intra-cluster and inter-cluster fusion. In the former, only data from *collaborator* nodes are fused, while in the latter, only data from *coordinator* nodes are fused or aggregated.

5.3.1. Intra-cluster fusion

Within the cluster, a shortest-path tree is formed so that each *collaborator* sends its data to the *coordinator* (tree root) using the shortest path composed only of *collaborator*

```
for all u ∈ S do
    role_u ← collaborator;
    Announce detection; {one broadcast}
    for all w ∈ N_u, such that w ≠ u do
        if id(u) < id(w) then
            role_u ← coordinator;
        end if
    end for
    if role_u = coordinator then
        C ← C ∪ {id(u)}; {C is the set of coordinators}
        Announce coordinator intention; {event-scoped flooding}
        Update C with other coordinator candidates;
    end if
    if id(u) = smallestid(C) then
        Announce coordinator condition; {net-scoped flooding}
    else
        role_u ← collaborator;
    end if
end for
```

Fig. 2. Cluster formation.

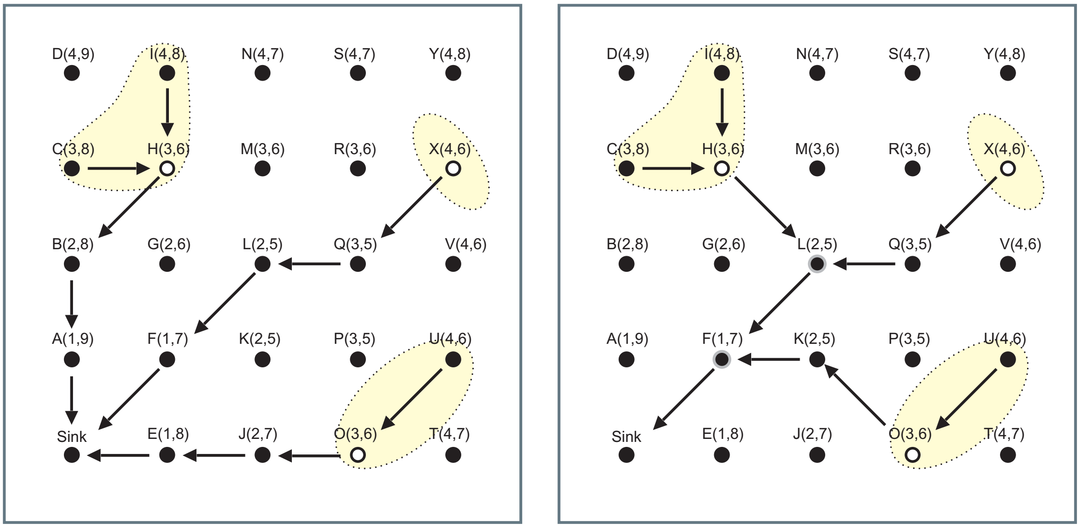
nodes. Then, *coordinator* nodes fuse data from the cluster members (*collaborator* nodes). When a *collaborator* is distant for more than one hop from the *coordinator*, *collaborator* nodes, which are used as intra-cluster relay, fuse or aggregate the packets being relayed. By doing this, regarding the number of resulting edges, the tree strategy used to connect *collaborators* to *coordinators* is not important. The reason is that the tree is composed only of *collaborators*, so each of them sends only one packet (multiple packets are aggregated) at every notification interval.

5.3.2. Inter-cluster fusion

Our algorithm looks for the shortest paths (connecting the cluster-heads to the sink node) that allow data aggregation of multiple clusters. For instance, consider the example depicted in Fig. 4. In this example, nodes H, O, and X are *coordinators* for three correlated events. The notation $a(b, c)$ used to label the nodes represents the distance table of each node, which means that node *a* is *b* hops from the sink and *c* is its aggregated coordinators-distance in hops.

```
for all u ∈ (V − S) do
    role_u ← relay;
    skdist_u ← ∞;
    for all w ∈ N_u, such that w ≠ u do
        test1_u ← dist-sk(w) < skdist_u;
        test2_u ← (dist-sk(w) = skdist_u) and (dist-co(w) < dist-co(nexthop_u));
        if test1_u or test2_u then
            nexthop_u ← w;
            skdist_u ← dist-sk(u);
        end if
    end for
    when node is ready to relay data do
        Aggregate and send all data to nexthop_u;
    end when
end for
```

Fig. 3. Route formation.



(a) Shortest path (b) InFRA routing

Fig. 4. Role assignment fusing multiple clusters: the notation $a(b,c)$ used to label the nodes represents the distance table of each node, which means that node a is b hops from the sink, and c is the sum of the distances (aggregated coordinators-distance) of node a to the current *coordinators*.

If the simple shortest path is used, we might have non-overlapping routes so that cluster data are not fused, as depicted in Fig. 4a. However, in our algorithm, we search for a tree that assures the shortest-path but enables the aggregation of data from multiple clusters, as depicted in Fig. 4b. In this example, InFRA is able to find a minimum shortest-path tree connecting all source nodes to the sink, in such a way that intra-cluster fusion is performed by nodes H, O, and X, and inter-cluster fusion is performed by nodes F and L.

5.4. Event detection

Event detection is assumed to be an information-fusion task [20]. Thus, by applying inference methods on the sensor observations, a sensor node can detect an event with an associated confidence measure [20]. This detection can be a collaborative or non-collaborative task depending on the available data, inference method, available resources and desired confidence. Since the event detection is not the focus of this work, to simplify or study, we assume that sensor nodes apply an inference method to detect events [20].

We also assume that events are described based on the events the network is designed to detect. For instance, let us assume our network detects fire only. Thus, we may consider the event location (fire sites) and area as the descriptor. In this case, if we have the same fire site detected by two disjoint clusters then we will have two fire sites (two neighbor fire sites), which is not really a problem from the application perspective. From the network perspective, in our algorithm, the events will be aggregated as soon as possible. If a set of nodes are in the same detection region of two neighbor fire sites, our algorithm will interpret as a single fire site. Again, this is not a critical issue since we will know where the fire (event) is. Clearly, there might be some scenarios, when the network detects multiple types of events, in which this approach may not lead to a good solution. However, we do not intend to provide a definitive solution for every case.

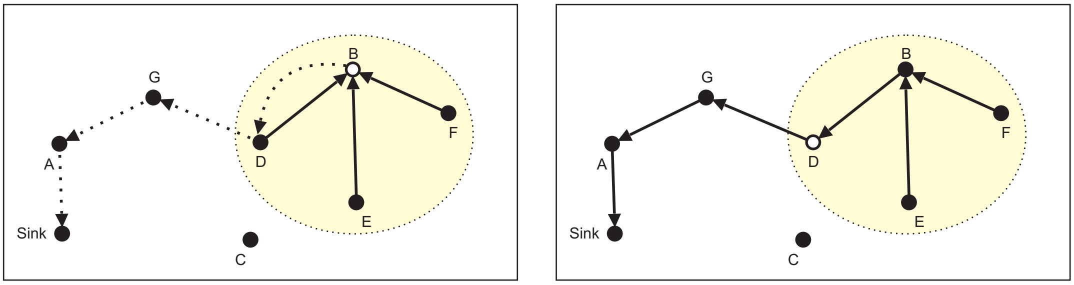
5.5. Dynamic topologies

In the proposed solution, dynamic topologies are addressed naturally. Reactive algorithms have the benefit of building the routing topology only when necessary, therefore, adapting themselves to topological changes that happen during the inactive periods. As we show latter on, the InFRA algorithm is of greater cost compared to other heuristics, but this cost is diluted if the duration of the event lasts long enough. Thus, what happens if a node fails during an event notification?

In the InFRA algorithm, if the chosen *relay* node fails, then the next packet is sent to the second best option, i.e., a new *relay* node is chosen. A node failure can be detected by the lack of activity, i.e., the packet is not forwarded, which is possible by taking advantage of the promiscuous nature of the wireless link. If the failure rate is so high that it compromises the routing information of every node (sink distance and aggregated coordinators-distance), then the InFRA algorithm may not find the best paths that it could. In fact, depending on the failure rate a proactive solution may be the best choice. For instance, when we consider the failure recovery problem, Nakamura et al. [21] shows that the periodic (proactive) topology rebuilding and a reactive solution may have the same behavior.

5.6. Role migration

In some cases, due to the strategy used for the *coordinator* election, a *collaborator* node may be chosen to relay its



(a) Initial role assignment: B is the coordinator

(b) After the role migration: D is the coordinator

Fig. 5. Role migration.

coordinator packets, which leads to waste of resources. To avoid such an undesirable situation, InFRA provides role migration function. Once a *collaborator* node identifies that it has to relay its *coordinator* packets, it assumes the *coordinator* role and broadcasts its condition to its neighbors. Then, all *collaborator* neighbors and the old *coordinator* send their data to the new *coordinator*. Nodes that are distant for more than one hop from the new *coordinator* will not be aware of the new cluster organization. However, this scenario does not lead to malfunctioning, because these nodes will send their data to the old *coordinator* that fuses them and forwards the result to the new *coordinator*.

Fig. 5 depicts the role-migration process. In the initial role assignment (Fig. 5a), node B becomes the *coordinator* and nodes D, E, and F become *collaborators*. However, after the intra-cluster fusion, node B sends its data towards the sink through the route B → D → G → A → Sink. This situation leads to waste of resources, since node D needs to send two packets every notification interval (one to node B and one to node G). When node D detects that it is relaying packets from its *coordinator*, it assumes that role and informs its neighbors. After that, all nodes send only one packet every notification interval (Fig. 5b).

6. Theoretical results

In this section, we present some theoretical results of the InFRA algorithm referring to the its efficacy in finding a solution to the minimum routing tree for event-driven WSNs, and discuss the feasibility of our solution compared to other heuristics.

For the sake of simplification, the theoretical analysis, presented in this section, and simulation evaluation, presented in Section 7, consider a lossless channel model. Such an assumption allows us to find the theoretical bounds that define some performance limits. To be fair, all algorithms were analyzed and evaluated under the same premise. A lossy channel will affect every aggregation solution. In general, algorithms that aggregate more data flows are more affected by lossy channels. The idea of our evaluation is to assess the infrastructure performance of the algorithms. When we have lossy channels, we can adopt some mechanism to overcome packet losses, such as retransmissions or data prediction (information fusion). The application requirements will determine whether or not it is worth implementing such mechanisms.

6.1. Approximation ratio

To derive analytical bounds for the approximation ratio of the InFRA algorithm, we use the concept of network diameter defined as follows.

We consider the network diameter D as the number of hops in the shortest path connecting the farthest node $v \in \mathbf{V}$ to the sink node.

Let us analyze the scenario in which our algorithm finds the worst solution compared to the optimal algorithm. Suppose we have $k \leqslant m$ groups of connected sources such that the shortest paths, between each of the groups and the sink node, have D hops and are disjointed (no route overlapping), and these k groups are separated by two hops from each other, as depicted in Fig. 6a. In this case, an optimal solution consists of a group reaching the sink node at D hops, the other $k-1$ groups reaching a neighbor group at two hops, as depicted in Fig. 6b. Plus, each of the remaining $m-k$ source nodes use one transmission link. In such a scenario, the communication cost of the optimal solution is

$$\mathrm{cost}(opt) = 2(k-1) + D + (m-k) = k + m + D - 2. \quad (5)$$

Although the InFRA algorithm tries to increase data aggregation, it prioritizes the shortest paths (see Section 5.2), which might lead to sub-optimal solutions. In the InFRA algorithm, each group of connected source nodes becomes a cluster. The $m-k$ *collaborators* use one transmission link, and the *coordinators* send their data through the disjointed shortest-paths. Thus, the communication cost of the InFRA solution is

$$\mathrm{cost}(infra) = kD + (m-k) = k(D-1) + m. \quad (6)$$

Eq. (6) represents the cost of the InFRA solution in the worst scenario – i.e., the worst case of the InFRA algorithm as a heuristic for the problem defined in Section 4 (a Steiner tree) — and Eq. (5) is the optimal cost in the same scenario. Thus, we can define the general approximation ratio of the InFRA algorithm as follows.

Theorem 6.1. *The approximation ratio of the InFRA algorithm is*

$$\text{cost}(infra) \leqslant \frac{k(D-1)+m}{k+D+m-2}\text{cost}(opt). \tag{7}$$

By exploiting the upper bound (7) in Theorem 6.1, we can determine the cases in which we surely obtain the optimal solution, and simpler bounds given certain conditions.

Theorem 6.2. *The approximation ratio of the InFRA algorithm decreases as k decreases in such a way that when* $k = 1, \text{cost}(infra) = \text{cost}(opt)$.

Proof. Looking at (7), we see that the contribution of k in the numerator proportional to $D-1$, while its contribution in the denominator is constant. Thus, the smaller the value of k, the smaller the result of (7). When we replace $k = 1$ in (7), we obtain $\text{cost}(infra) = \text{cost}(opt)$. □

Theorem 6.3. *Compared to the optimal solution, the InFRA solution uses no more than* $O(k)$ *additional hops.*

Proof. By subtracting (5) from (6), we obtain

$$\begin{aligned}
\text{cost}(infra) - \text{cost}(opt) &= k(D-1) + m - (k+m+D-2) \\
&= kD - k + m - k - m - D + 2 \\
&= kD - 2k - D + 2 \\
&= (D-2)k - (D-2) \\
&= (D-2)(k-1).
\end{aligned}$$

Since D remains constant for a given network, we have $\text{cost}(infra) \leqslant \text{cost}(opt) + O(k)$. □

This result means that the number of extra edges in the InFRA solution is at most $O(k)$, where k is the number of clusters formed by the InFRA algorithm.

Corollary 6.3.1. *If for every* $u \in \mathbf{S}$ *exists a* $v \in \mathbf{S}$ *such that* $\langle u, v\rangle \in E$*, then the minimum routing tree is not NP-complete, and* $\text{cost}(infra) = \text{cost}(opt)$.

Proof. When such a hypothesis holds, i.e., for all $u \in \mathbf{S}$ exists a $v \in \mathbf{S}$ such that $\langle u, v\rangle \in E$, then the InFRA algorithm builds only one cluster, i.e., $k = 1$. Therefore, according to Theorem 6.2 $\text{cost}(infra) = \text{cost}(opt)$. □

Corollary 6.3.1 shows that when the network is detecting only one event, which is reasonable for several applications, the InFRA algorithm finds the optimal solution.

Theorem 6.4. *The InFRA algorithm always finds the optimal solution when* $D \leqslant 2$.

Proof. When $D = 1$, every *coordinator* node sends its packets directly to the sink node, i.e., no *relay* node is used. When $D = 2$, by replacing the D value in (7), we obtain $\text{cost}(infra) = \text{cost}(opt)$. □

Theorem 6.5. *When* $D > 2$*, the approximation ratio of the InFRA algorithm is limited by* $D-1$*, i.e.,*

$$\frac{\text{cost}(infra)}{\text{cost}(opt)} \leqslant \frac{k(D-1)+m}{k+m+D-2} < D-1. \tag{8}$$

Proof. If we develop inequality (8), we obtain

$$\begin{aligned}
&k(D-1) + m < (D-1)(k+m+D-2) \\
&k(D-1) + m < (D-1)k + (D-1)m + (D-1)(D-2) \\
&m < (D-1)m + (D-1)(D-2). \qquad (9)
\end{aligned}$$

Since $D > 2$, we have $(D-1) > 1$ and $(D-2) > 0$. Thus, (9) is true, which means that (8) is also true, i.e., $\text{cost}(infra) < (D-1) \times \text{cost}(opt)$. □

Corollary 6.5.1. *When the network has a constant diameter, the approximation ratio of the InFRA algorithm is* $O(1)$.

Proof. When D is constant, according to Theorem 6.5, the approximation ratio is $O(1)$. □

This result shows that if a network does not increase its diameter – which is a reasonable assumption for many applications – despite the number of event detections m, the InFRA algorithm has a $O(1)$-approximation ratio.

Theorem 6.5 shows that $\text{cost}(infra) < (D-1)\times \text{cost}(opt)$, which in some cases means that $\text{cost}(infra) < O(1) \times \text{cost}(opt)$ (see Corollary 6.5.1). However, in large-scale networks, D can be too large. Therefore, we must have a better approximation ratio for such cases.

Theorem 6.6. *When* $D \to \infty$*, the approximation ratio of the InFRA algorithm is k.*

Proof. When $D \to \infty$, we have:

$$\frac{\text{cost}(infra)}{\text{cost}(opt)} \leqslant \lim_{D\to\infty} \frac{k(D-1)+m}{k+D+m-2} = \frac{kD}{D} = k.$$

Therefore, $\text{cost}(infra) \leqslant k \times \text{cost}(opt)$. □

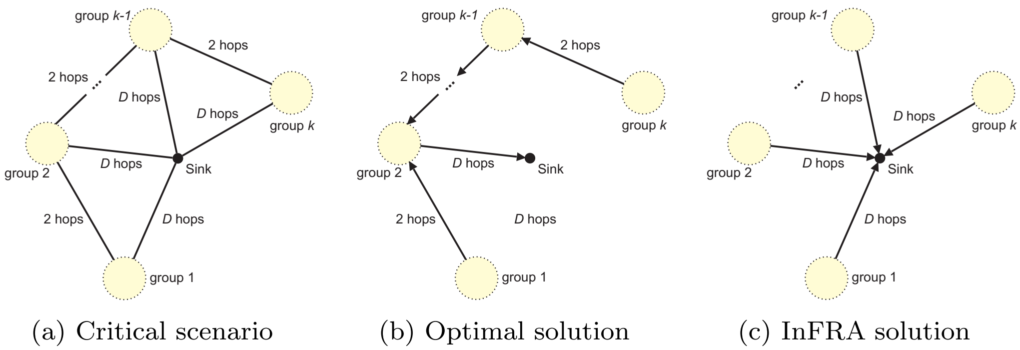


Fig. 6. Scenario in which the InFRA algorithm retrieves the worst solution.

When $m/D \to 0$, we have the same behavior as $D \to \infty$. Thus, in this case, $\text{cost}(infra) < k \times \text{cost}(opt)$ as well.

Corollary 6.6.1. *When no source node does not have a source node as a neighbor, InFRA algorithm has an $O(1)$-approximation ratio.*

Proof. When no source node does not have a neighbor source node, InFRA finds $k = m$

$$\frac{\text{cost}(infra)}{\text{cost}(opt)} \leqslant \frac{m(D-1)+m}{m+D+m-2} = \frac{mD}{2m+D-2} = O(1). \qquad \square$$

Corollary 6.6.1 is an interesting result for scenarios in which the event radius is so small that it is detected by only one source node. Although these scenarios might not be frequent, they might occur, especially when density control is performed.

6.2. A complexity analysis

In this section, we compare the communication complexity of the InFRA algorithm with other heuristics for the Steiner tree problem.

The best known heuristic for the Steiner Tree has a 1.55-approximation ratio [9]. However, this heuristic is not suitable for distributed implementation. The best distributed algorithm that we know of is the Greedy Incremental Tree (GIT) [15] that has a 2-approximation ratio [7]. In the GIT heuristic, the tree initially consists of the shortest path between the sink and the nearest source, and at each step after that the source closest to the current tree is connected to the tree.

Although the GIT heuristic is able to find good approximations, its distributed version [8] presents severe limitations for WSNs. First, all nodes necessarily have to know their shortest paths to the other nodes in the network. The communication cost for obtaining such information is $O(n^2)$, because every node needs to flood its location. Second, the memory space to store those paths locally (at each sensor node) is $O(D \times n)$, because the maximum route can have D hops (the network diameter). Once these shortest paths are available, the algorithm takes $O(mn)$ messages to build the routing tree [8]. These costs are not affordable for large-scale networks composed of limited-memory sensor nodes.

As we demonstrate in Theorem 6.6, the approximation ratio of the InFRA algorithm can be $k \times \text{cost}(opt)$ for large-scale networks. When $k > 2$, this is clearly worse than the GIT's approximation ratio. However, the InFRA algorithm takes $O(m)$ transmissions to create the clusters and $O(kn)$ to flood the clusters' information. In addition, each node maintains a routing table with an entry for each neighbor, and each entry contains only the node *id*, the coordinators-aggregated distance, and the sink distance referring to that neighbor.

As Woo et al. [22] show, static trees are very susceptible to the lossy nature of wireless links. Thus, another drawback of the GIT heuristic is that the algorithm needs to be executed every time a node in the routing tree fails, which demands $O(mn)$ messages per failure. On the other hand, in the InFRA heuristic, each node chooses its parent only when a packet is available; if the best node fails, the second best node is chosen without additional communication cost.

7. Simulation results

The simulation experiments of the InFRA algorithm use the Sinalgo simulator [23]. In all graphs, the curves represent the average values, while the error bars represent the confidence intervals for 95% confidence for 100 different instances (seeds). The simulation parameters are based on the MicaZ sensor node [24], which uses the 802.15.4 standard. The default parameters for the experiments are presented in Table 1.

7.1. Methodology

The experiments compare InFRA with one proactive solution and two reactive solutions. For the sake of simplicity, to represent the proactive role assignment we use the a periodic shortest-path tree (PSPT) [10]., i.e., a SPT that is periodically built. This is a simple and popular solution for delivering data to the sink node. In this strategy, the sink node periodically broadcasts a building packet. Each node chooses the candidate closer to the sink as its parent node and forwards the building packet.

For the reactive candidate we choose a reactive variant of the PSPT algorithm, or simply SPT, and the reactive Centered-at-Nearest-Source tree (CNS) [10]. In the SPT algorithm, when an event is detected, the source nodes flood a notification packet to the sink node. When the sink receives that packet it triggers the building process used by the SPT strategy. In the CNS aggregation scheme, all sources send their data to the source nearest to the sink. Then, this source sends the aggregated information to the sink through the shortest path. However, once an event is detected a flooding is performed to announce the event detection and another is performed to build the tree.

We evaluate the algorithms using the metrics:

- Data packets – total number of data packet transmissions in the network. It shows how well the algorithms are relaying the data packets.
- Packet overhead – total number of control packet transmissions in the network. It shows the cost to assign roles for event notification.
- Routing efficiency – total number of packet transmissions used to process and deliver all data packets generated by source nodes. It is measured in packets per data processed.

In all experiments, the delivery ratio was greater than 95% for all algorithms, therefore, we decided not to show the graphs for the success ratio. The reason for such a high delivery ratio is that we did not introduce any failures in our simulations.

Because of the difficulty to compute the optimal solution we define the lower bound cost of a routing tree in the same way as Krishanamachari et al. [10], which is composed of the shortest path between the source closest to the sink plus one hop for each remaining sources:

Table 1
Default scenario configuration.

Parameter	Value
Sensor field	$700 \times 700\ m^2$
Sink nodes	1 (bottom left)
Size	529 nodes (disturbed grid)
Communication range	50 m
Bandwidth	250 kbps
Simultaneous events	1 (top right)
Event radius	80 m
Event duration	3 h
Inactivity time	2000 s
Notification rate	60 s

$$\mathrm{cost}(lowerbound) = \mathrm{shortest} - \mathrm{path}(u, v_1) + (m - 1), \tag{10}$$

where $u \in \mathbf{S}$ is the source closest to the sink v_1 and $m = |\mathbf{S}|$ is the number of source nodes.

In the simulation scenarios with only one event, (10) represents the optimal solution, and in scenarios with multiple events we have $\mathrm{cost}(opt) \geqslant \mathrm{cost}(lowerbound)$. In the following sections, data packets graphics show the theoretical lower bound.

7.2. Reactive vs. proactive role assignment

In a proactive approach, roles are assigned even when no event is being detected. Thus, such a role assignment should be executed periodically to recover from topological changes (e.g., node failures). In reactive strategies, roles are assigned only when an event is detected avoiding the need for periodical executions. Thus, it is not fair to compare a proactive role assignment with a reactive one because in the reactive case the algorithm will easily outperform proactive strategies if the network is inactive for a long time. To illustrate our viewpoint, we simulated a 529-node network in a $700 \times 700\ m^2$ field. We placed the sink node in the bottom left corner of the sensor field and generated one event with a 80 m radius in the opposite corner. This event starts at 1000 s and lasts 10,800 s. For the proactive role assignment, we choose the PSPT with opportunistic data aggregation. In the PSPT strategy, roles are reassigned every 200 s to build the routing (shortest-path) tree.

Fig. 7 depicts the behavior for the first 900 s of simulation. In this graph, the vertical axis represents the total amount of packets sent by all nodes in the network. Clearly, the proactive strategy sends more packets because it rebuilds the tree periodically. However, when we compare InFRA with SPT, we can see that although in the InFRA strategy more packets are used in the role assignment phase, data aggregation is enhanced in such a way that if the event remains active long enough, the initial overhead is compensated by the savings due to data aggregation.

In the next experiments, we compare the InFRA algorithm only to the reactive algorithms, since we could easily find a scenario in which the reactive solutions outperform the proactive PSPT.

7.3. Communication range

Here, we evaluate the impact of the communication range in the algorithms by changing the communication range from 50 m to 100 m (maximum range for the micaZ sensor node). The results are shown in Fig. 8.

As a result, we the communication range increases, the node degree increases accordingly (Fig. 8d), which means that routes are smaller because the number of nodes and the area of the sensor field remain constant (density is constant). Consequently, the three algorithms send less data packets when the communication range increases (Fig. 8a). However, InFRA sends less packets due to better data aggregation. Particularly, when the communication range is 50 m, SPT and CNS algorithms use nearly 45% more packets to deliver the sensed data.

Fig. 8b shows that, independent of the communication range, SPT and CNS always have the same overhead to assign the relay/aggregation roles. However, the InFRA overhead decreases as the communication range increases. The reason is that a communication range smaller than the event radius (80 m) increases the probability of multihop communication within the clusters. As a result, the probability of multiple *coordinator* candidates is greater, and occasionally one *coordinator* candidate may not receive the notification of another candidate. Consequently, we may have two or more *coordinators* (per event) flooding their condition. This experiment shows how packet losses can affect a Steiner-tree heuristic algorithm. When the relation communication range is large enough (100 m), the InFRA algorithm finds the optimal solution, but this situation changes as the communication range gets smaller (Fig. 8a). Despite this fact, because InFRA aggregates more packets, it still uses the communication resources more efficiently than SPT and CNS (Fig. 8c).

7.4. Network scalability

To evaluate the network scalability we increase the network size from 121 to 1024 nodes, and re-size the sensor field to keep a constant network density of 8.48. We consider the network density as the relation $n\pi r^2/A$, where n is the number of nodes, r is the communication range, and A is the area of the sensor field. The objective of keeping a constant network density is to isolate the scale influ-

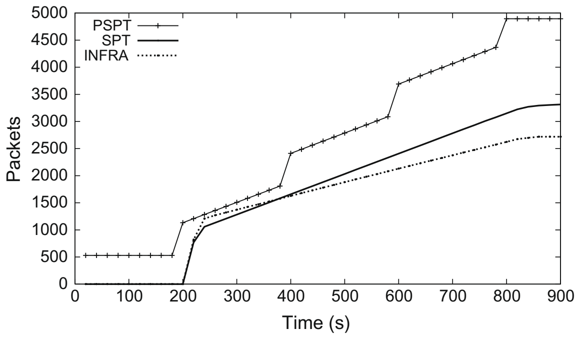


Fig. 7. Packet transmissions along time.

ence by keeping a constant node degree and a constant number of sources (Fig. 9d).

Fig. 9 shows that the InFRA is more scalable than the other algorithms. The reason is that the InFRA reduces the data transmissions by increasing data aggregation (Fig. 9a). However, during the role assignment phase, the InFRA sends more packets than the SPT and the CNS because the source nodes perform a *coordinator* election (Fig. 9b). The most important result is that, as the network size increases, the InFRA uses less packets to process and deliver the data packets generated by source nodes (Fig. 9c). Particularly, when the network size is 1024, even though InFRA has greater overhead, it uses nearly 70% of the communication resources used by SPT.

7.5. Event scalability

To evaluate how the algorithms behave when the number of simultaneous events increases, we simulate 529-node networks, increasing the number of simultaneous events from 2 to 6. In this particular case, the lower bound is not necessarily the optimal solution, as we mentioned before, for multiple events cost(opt) $\geqslant$ cost($lowerbound$).

Obliviously, the number of data packet transmissions (Fig. 10a) and the assignment overhead (Fig. 10b) increase with the number of simultaneous events, because the amount of source nodes increases as well (Fig. 10d). However, as Fig. 10c shows, the routing efficiency improves since more data is aggregated. As the number of events increases, the difference in the routing efficiency of InFRA and SPT tends to decrease slightly. The reason is that the probability of route overlapping increases. Particularly, the CNS strategy performs poorly with simultaneous events, because all source nodes send their data to the source closest to the sink, reducing the data aggregation ratio. In addition, the CNS strategy eventually uses longer routes than SPT and InFRA, which aggravates its poor performance. Particularly, Fig. 10c shows that even though InFRA has greater overhead, it spends up to 70% of the communication resources used by CNS in all simulated scenarios.

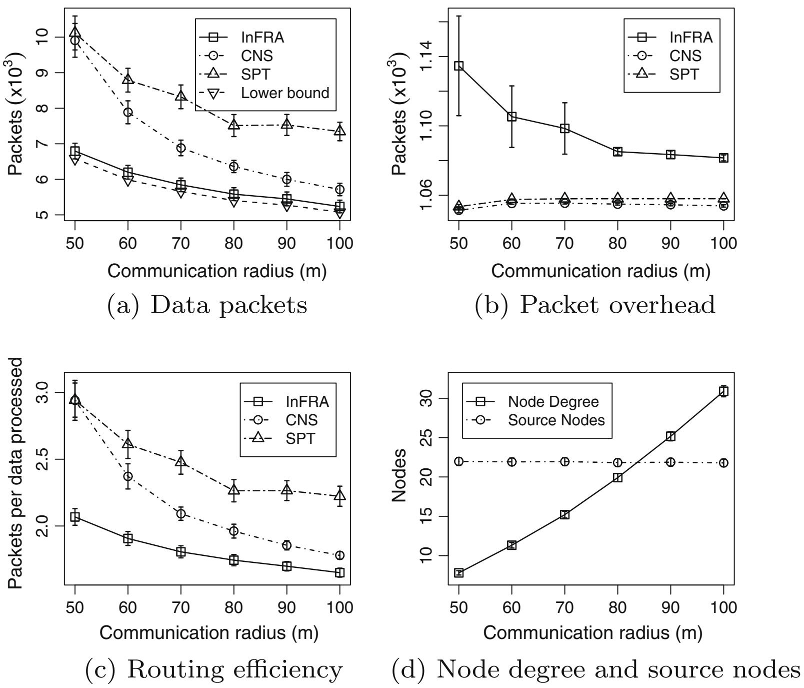
7.6. Event size

We also evaluated the impact of the event size, i.e., the influence region in which a sensor node can detect an event. This is accomplished by increasing the event radius from 50 m to 100 m and keeping the communication range fixed at 80 m. The results are shown in Fig. 11.

As a general result, InFRA outperforms SPT and CNS by reducing the number of data packet transmissions (Fig. 11a) and, consequently, using the communication resources more efficiently (Fig. 11c). However, in this evaluation we stress that, when the relation between the event radius and communication range increases, the overhead introduced by InFRA also increases (Fig. 11b). The reason is that eventually we have multiple *coordinator* candidates per event, which occasionally results in multiple *coordinators* per event, especially when the number of source nodes per event increases (Fig. 11d).

7.7. Density

Although network and source density where isolatedly evaluated in Section 7.3 and Section 7.6, respectively, in



(a) Data packets (b) Packet overhead

(c) Routing efficiency (d) Node degree and source nodes

Fig. 8. Communication range.

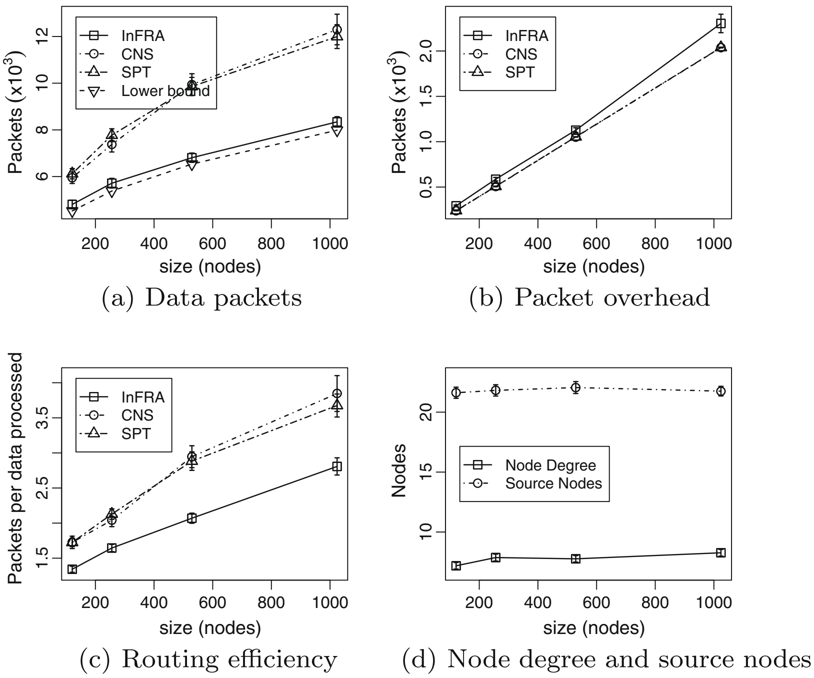


Fig. 9. Network scalability.

this section we evaluate the density impact by keeping the sensor field constant and vary the number of sensor nodes from 529 to 4096. By doing this, both the number of nodes and the number of sources increase (Fig. 12d), therefore, increasing network and source densities (since the sensor field has a constant area). In this case, InFRA outperforms SPT and CNS, as Fig. 12a–c show. Again, as the network density increases, SPT and CNS present the same behavior, but the InFRA algorithm remains more efficient (Fig. 12c).

7.8. Event duration

To determine the conditions when InFRA is worth adopting, and when a simpler solution, such as SPT, should

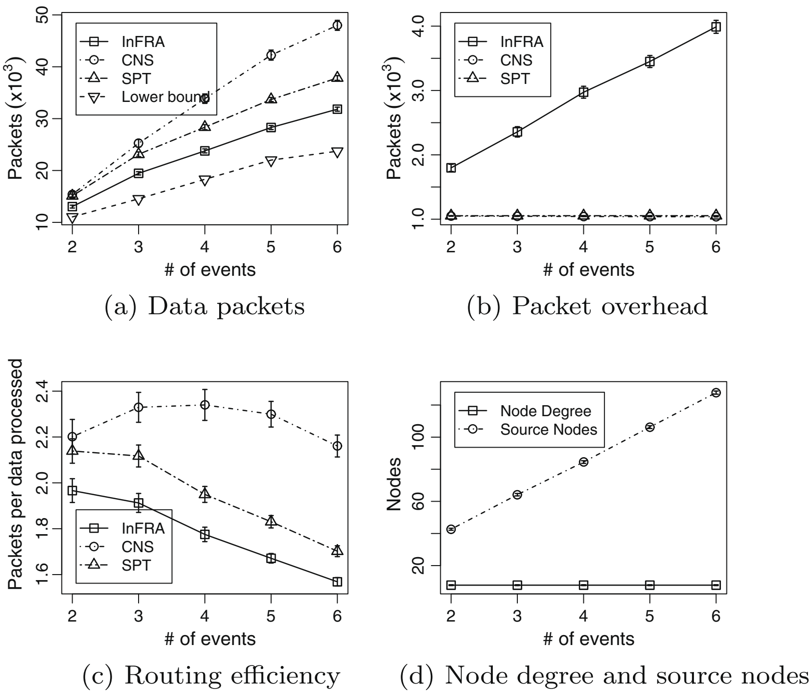
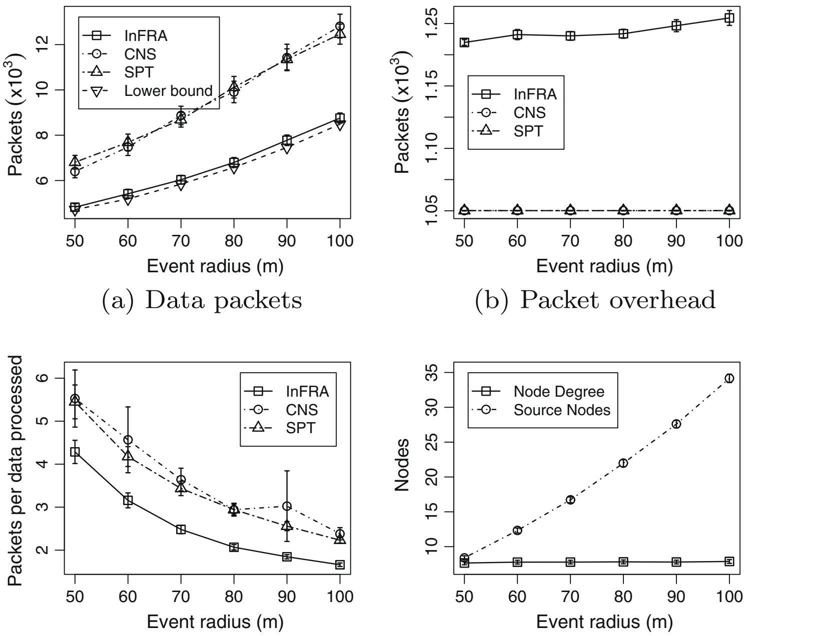


Fig. 10. Event scalability.



(a) Data packets (b) Packet overhead

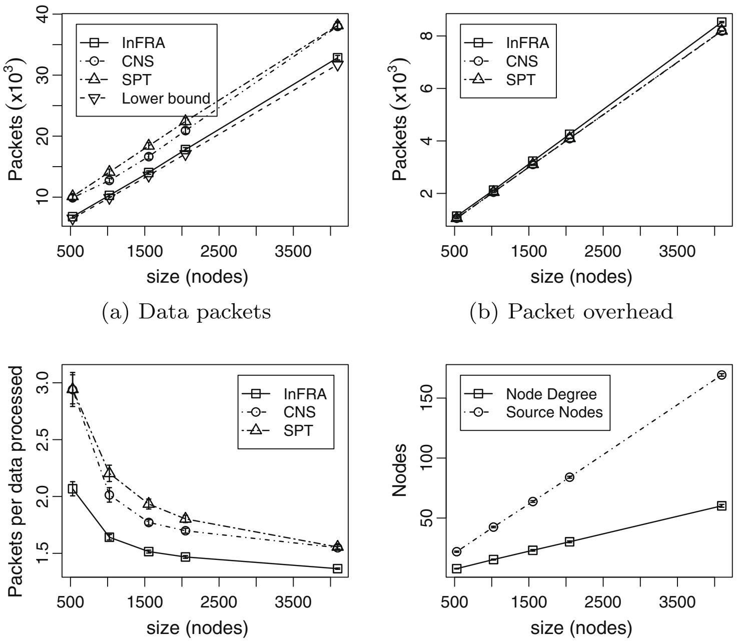
(c) Energy efficiency (d) Node degree and source nodes

Fig. 11. Event size.

be chosen, we varied the event duration from 0.5 h to 4 h, and kept the other default parameters. In this case, we can state that when the events last for less than 2 h, the simpler SPT outperforms InFRA and CNS. The reason is that although InFRA aggregates more packets (Fig. 13a) it presents a greater overhead (Fig. 13b), and if the event is too short (smaller then 2*h*) the gains with additional data aggregation does not pay for the overhead required to build the routing tree (Fig. 13c).

8. Final remarks

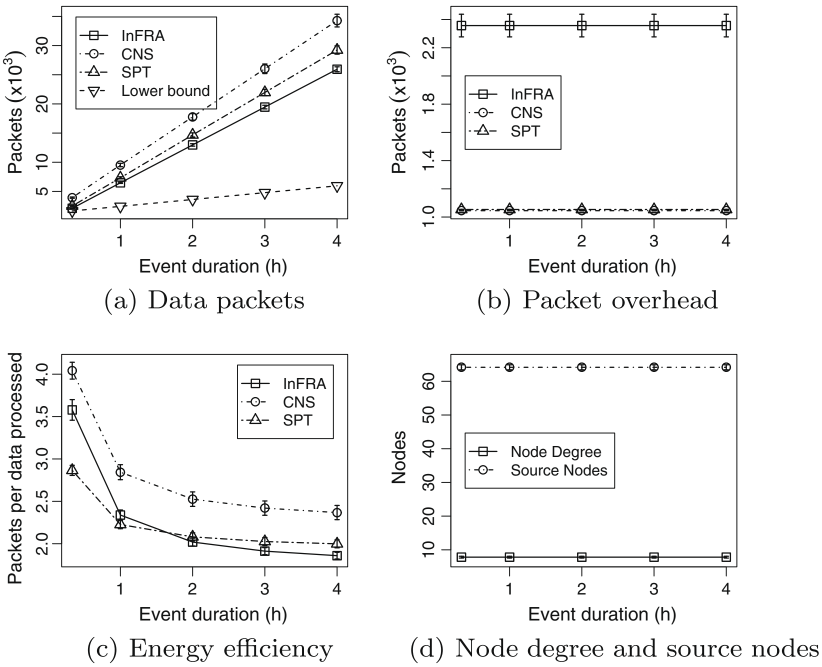
In this paper, we formalize a role assignment model and propose a reactive algorithm, called InFRA, that starts the



(a) Data packets (b) Packet overhead

(c) Energy efficiency (d) Node degree and source nodes

Fig. 12. Network and source density.



(a) Data packets (b) Packet overhead

(c) Energy efficiency (d) Node degree and source nodes

Fig. 13. Event duration.

assignment process only when an event is detected, therefore, saving energy during periods of inactivity. The objective of this algorithm is to build a routing infrastructure for delivering data to the sink and increase the data aggregation probability. The proposed scheme is a distributed heuristic to find a minimum Steiner tree connecting source nodes to the sink.

Our theoretical results show that this heuristic has $O(1)$-approximation ratio when the network diameter remains constant and, in large-scale networks, it has a k-approximation ratio.

Our simulation evaluation compares the InFRA algorithm with reactive versions of the shortest-path tree (SPT) and the centered-at-nearest-source tree (CNS). This evaluation covers the assessment of different factors: network scalability, event scalability, communication range, and event size. The results show that although the InFRA algorithm presents a higher overhead, it outperforms SPT and CNS by finding routes of higher data aggregation ratios. In some cases, the InFRA algorithm uses only 70% of the communication resources required by SPT, and for multiple events it uses only 70% of the communication resources required by CNS.

We also show that the InFRA strategy is worth using when we expect the events to last for some hours ($> 2h$), which is very reasonable for events such as fire that usually take a few hours to be controlled. However, for applications with events that last a few minutes or seconds, a simpler solution such as SPT is a better option.

The presented evaluation comprehends static events of fixed radius. We plan to work on the assessment of InFRA when events present dynamic sizes (events of increasing and decreasing sizes) and can move across the sensor field.

Acknowledgement

This work is partially supported by the National Council for Scientific and Technological Development (CNPq), an agency linked to the Brazilian Ministry of Science and Technology (MCT), under the Grant Numbers 47.4194/2007-8 (RastroAM) and 48.1069/2004-6 (Funde).

References

[1] Eduardo F. Nakamura, Horacio A.B.F. de Oliveira, Luciana F. Pontello, Antonio A.F. Loureiro, On demand role assignment for event detection in sensor networks, in: Proceedings of the 11th IEEE International Symposium on Computers and Communication (ISCC'06), Cagliari, Italy, June 2006, pp. 941–947.
[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422.
[3] Manish Bhardwaj, Anantha P. Chandakasan, Bounding the lifetime of sensor networks via optimal role assignment, in: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), New York, NY, USA, June 2002.
[4] Boris Jan Bonfils, Philippe Bonnet, Adaptive and decentralized operator placement for in-network query processing, in: Feng Zhao, Leonidas J. Guibas (Eds.), IPSN'03, Lecture Notes in Computer Science, vol. 2634, Palo Alto, USA, April 2003, pp. 47–62.
[5] Rajnish Kumar, Matthew Wolenetz, Bikash Agarwalla, JunSuk Shin, Phillip Hutto, Arnab Paul, Umakishore Ramachandran, Dfuse: a framework for distributed data fusion, in: [25], pp. 114–125.
[6] Manish Kochhal, Loren Schwiebert, Sandeep Gupta, Role-based hierarchical self organization for wireless ad hoc sensor networks, in: Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA'03), San Diego, CA, USA, September 2003, pp. 98–107.
[7] H. Takahashi, A. Matsuyama, An approximate solution for the Steiner problem in graphs, Mathematica Japonica 24 (6) (1980) 573–577.
[8] Fred Bauer, Anujan Varma, Distributed algorithms for multicast path setup in data networks, Transactions on Networking 4 (2) (1996) 181–191.

[9] Gabriel Robins, Alexander Zelikovsky. Improved Steiner tree approximation in graphs, in: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'00), San Francisco, USA, 2000, pp. 770–779, ISBN:0-89871-453-2.
[10] B. Krishanamachari, D. Estrin, S. Wicker, The impact of data aggregation in wireless sensor networks, in International Workshop of Distributed Event Based Systems, Vienna, Austria, July 2002, pp. 575–578.
[11] Koustuv Dasgupta, Meghna Kukreja, Konstantinos Kalpakis, Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks, in: Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC'03), Antalya, Turkey, vol. 1, June/July 2003, pp. 341–348.
[12] M. Kochhal, L. Schwiebert, S. Gupta, Integrating sensing perspectives for better self organization of ad hoc wireless sensor networks, Journal of Information Science and Engineering 20 (3) (2004) 449–475.
[13] Christian Frank, Kay Römer, Algorithms for generic role assignment in wireless sensor networks, in: Jason Redi, Hari Balakrishnan, Feng Zhao (Eds.), SenSys'05, San Diego, USA, November 2005, pp. 230–242.
[14] Razvan Cristescu, Baltasar Beferull-Lozano, Martin Vetterli, On network correlated data gathering, in: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04), Hong Kong, IEEE, March 2004, pp. 2571–2582.
[15] Bhaskar Krishnamachari, Deborah Estrin, Stephen Wicker, The impact of data aggregation in wireless sensor networks, in: International Workshop of Distributed Event Based Systems (DEBS), Vienna, Austria, July 2002, pp. 575–578.
[16] Pietro Ciciriello, Luca Mottola, Gian Pietro Picco, Efficient routing from multiple sources to multiple sinks in wireless sensor networks, in: Koen Langendoen, Thiemo Voigt (Eds.), Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN'07), Lecture Notes in Computer Science, vol. 4373, Springer, Delft, Netherlands, January 2007, pp. 34–50.
[17] Kai-Wei Fan, Sha Liu, Prasun Sinha. Scalable data aggregation for dynamic events in sensor networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys'06), New York, NY, USA, ACM, 2006, pp. 181–194. ISBN:1-59593-343-3.
[18] Horacio A.B.F. Oliveira, Eduardo F. Nakamura, Antonio A.F. Loureiro, Azzedine Boukerche, Error analysis of localization systems in sensor networks, in: Proceedings of the 13th ACM International Symposium on Geographic Information Systems (GIS'05), Bremen, Germany, ACM, November 2005, pp. 71–78.
[19] F. Hwang, D. Richards, P. Winter, The Steiner Problem, Annals of Discrete Mathematics, North-Holland, 1992.
[20] Eduardo F. Nakamura, Antonio A.F. Loureiro, Alejandro C. Frery, Information fusion for wireless sensor networks: methods, models, and classifications, ACM Computing Surveys 39 (3) (2007). A9/1–A9/55.
[21] Eduardo F. Nakamura, Carlos M.S. Figueiredo, Fabiola G. Nakamura, Antonio A.F. Loureiro, Diffuse: a topology building engine for wireless sensor networks, Signal Processing 87 (12) (2007) 2991–3009.
[22] Alec Woo, Terence Tong, David Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in: [25], pp. 14–27.
[23] Distributed Computing Group ETH Zurich, Sinalgo – Simulator for network algorithms, Access: January 2008 (Online). <http://dcg.ethz.ch/projects/sinalgo/>.
[24] Crossbow, Micaz - wireless measurement system, 2007. <http://www.xbow.com/>.
[25] Ian F. Akyildiz, Deborah Estrin, David E. Culler, Mani B. Srivastava (Eds.), Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03), Los Angeles, USA, November 2003.

Eduardo Freire Nakamura is a Researcher and Professor at the Research and Technological Innovation Center, Brazil. He received his Ph.D. in Computer Science from the Federal University of Minas Gerais, Brazil, in 2007. His research interests include data/information fusion, distributed algorithms, localization algorithms, wireless ad hoc and sensor networks, mobile and pervasive computing. In the last 7 years, he has published regularly in international conferences and journals related to those areas, and also presented tutorials at international conferences, such as ACM SIGMOD and IEEE CIT. He has also served as a TPC member for international conferences, such as, PIMRC, ICCCN, WCNC, AINA, WAC, and LAACS.

Heitor S. Ramos received the MS degree in Computing Modeling from Federal University of Alagoas, Brazil, in 2006. He is currently working toward the Ph.D. degree in Computer Science at the Federal University of Minas Gerais, Brazil. His research interests include mathematics models in wireless sensor networks, distributed algorithms, mobile and pervasive computing.

Leandro Aparecido Villas is a Ph.D. student of Computer Science at the Federal University of Minas Gerais, Brazil. He received his Master Degree in Computer Science from the Federal University of Sao Carlos, Brazil, in 2007. His research interests include context interpretation, distributed algorithms, routing algorithms, wireless actor and sensor networks, quality of Service. He has published papers in the area of wireless sensor networks and received the prize for best article of the XXV Brazilian Symposium on Computer Networks and Distributed Systems, SBRC.

Horacio A.B.F. Oliveira is an Associate Professor of Computer Science at Federal University of Amazonas (UFAM), Brazil. He received his Ph.D. in Computer Science from the Federal University of Minas Gerais, Brazil with a partial doctoral fellowship at University of Ottawa, Canada, in 2008. His research interests include localization and synchronization algorithms, distributed algorithms, and wireless ad hoc, vehicular, and sensor networks. He is author of several papers in the different areas of his research interests.

André Luiz Lins de Aquino is Professor at Federal University of Ouro Preto, Brazil. He received his Ph.D. in Computer Science from the Federal University of Minas Gerais, Brazil, in 2008. His research interests include data reduction, distributed algorithms, wireless ad hoc and sensor networks, mobile and pervasive computing. In addition, he has published several papers in the area of wireless sensor networks.

Antonio Loureiro received his B.Sc. and M.Sc. degrees in computer science from the Federal University of Minas Gerais (UFMG), Brazil, and the Ph.D. degree in computer science from the University of British Columbia, Canada. Currently, he is a professor of computer science at UFMG, where he leads the research group in wireless sensor networks. His main research areas are wireless sensor networks, mobile computing, and distributed algorithms. In the last 10 years he has published regularly in international conferences and journals related to those areas, and also presented tutorials at international conferences.