

# An Integrated Approach for Density Control and Routing in Wireless Sensor Networks

Isabela G. Siqueira<sup>1</sup>, Carlos Maurício S. Figueiredo<sup>1,2</sup>,  
Antonio Alfredo F. Loureiro<sup>1</sup>, José Marcos Nogueira<sup>1</sup>,  
and Linnyer Beatrys Ruiz<sup>1</sup>

<sup>1</sup> Federal University of Minas Gerais  
Dept. of Computer Science  
Belo Horizonte, MG, Brazil

{isabela,mauricio,loureiro,jmarcos,linnyer}@dcc.ufmg.br

<sup>2</sup> FUCAPI – Analysis, Research and  
Technological Innovation Center  
Manaus, AM, Brazil  
mauricio.figueiredo@fucapi.br

## Abstract

*Wireless Sensor Networks (WSNs) are characterized by having scarce resources. The usual way of designing network functions is to consider them isolatedly, a strategy which may not guarantee the correct and efficient operation of WSNs. For this reason, in this paper we propose an integrated design of network functions. We take two important WSN functions — density control and routing — as an example and present two approaches to integrate them. In particular, we present two solutions, named RDC-Sync and RDC-Integrated, which integrate a geographical density control algorithm with tree routing. The simulations experiments performed prove that the integrated design improves the network performance, especially when density control and routing are fully integrated.*

## 1. Introduction

Wireless Sensor Networks (WSNs) are composed by compact and low-cost devices, which are capable of sensing, computing, and communicating through wireless links. When in a great number, the elements of these networks, the sensor nodes, are capable of performing important sensing tasks through collaboration. It is predicted that WSNs will be the key to support several applications in the future, such as traffic monitoring, disaster recovery, intrusion detection, and military applications [5].

The major challenge to research in WSNs lies in the elaboration of solutions that make efficient use of the network resources. Unlike the devices of traditional

networks, sensor nodes have very low capacity, mainly regarding their energy stock (battery). Unfortunately, recharging or replacing batteries may be inconvenient, or even unfeasible, in case of deployment in regions of hard or impossible physical access. For these reasons, energy-efficiency must be a primary aspect of every solution for WSNs.

The fundamental objective of WSNs is to report information from the environment to the network observer. In order to accomplish this task, sensor nodes need to deliver data collected from the environment to an external entity — a “sink node” — which will process and analyze them. Since the sensor nodes normally do not have enough power to transfer data directly to the sink, routing is a fundamental function for successful delivery of data. But for this delivery to be efficient, it is necessary that the network should adopt mechanisms for controlling its density. Normally, this control is possible through scheduling of the sensor nodes activities. The aim is to keep only a minimum set of active sensor nodes at each period. Nodes which are not chosen to be part of this set can turn off their devices (radio, sensor, etc.) and remain in a “sleep mode” until they are required to come back to activity. Besides providing the extension of the network lifetime, this strategy reduces the network traffic and facilitates the routing job, since the amount of data produced becomes lower.

With the purpose of guaranteeing a correct and efficient data delivery, to introduce these two important functions — routing and density control — in a WSN design is a natural approach. Nevertheless, it is essential that they should be combined carefully. For instance, since the density control makes changes to the network topology, if there is a routing infrastructure, it

can be affected by route breaks, which may lead to data lost. Sometimes, it may be better to have these functions performing in cooperation, instead of separately; for that, an integrated design of them is recommended. If this is the case, these functions will know each other, sharing information and performing their operations in a synchronized manner (for instance, one after the other). This allows the network to operate more correct and efficiently. If the integration approach adopted is to combine the purposes of routing and density control in a single solution, even better results can be achieved.

In this paper we suggest the integrated design of routing and density control in the development of WSNs solutions. This kind of design in general, known as “cross-layer design”, introduces a new level in the research for better solutions for wireless networks. In the context of WSNs, specifically, many approaches already exist, mostly including only aspects of communication protocols (mainly MAC and routing). As far as we know, there is no solution available that integrates density control and routing. In this paper, we provide this novel vision for WSNs.

Our main contributions are: (1) analysis of impacting factors of density control over routing; (2) presentation of an integration strategy which considers both density control and routing tasks independently, but synchronized; (3) presentation of an approach to fully integrate particular density control and routing algorithms inside a single solution; and (4) presentation of two solutions (RDC-Sync and RDC-Integrated) based on two distinct strategies to integrate a traditional tree routing solution for WSNs (following the EF-Tree [6] implementation) and a density control algorithm called OGDC [19].

With the purpose of evaluating the performance of the integrated solutions and hence showing the importance of routing and density control integrated design, simulation experiments were conducted. The results prove that the integration of these two essential functions is a step further on providing efficient solutions for WSNs. If their combination is not carefully designed, the benefit of controlling the network redundancy can be wasted, mainly because of unsought information losses. In particular, the results show that the fully integrated solution (RDC-Integrated) reduces the network overhead and energy consumption, while significantly improving the amount of information collected by the sensor nodes which is correctly delivered to the network observer.

This document is organized as follows. In Section 2, the related work is presented. In Section 3, the aspects of density control that may influence the performance of routing are analyzed. In Section 4, the integrated

approaches (RDC-Sync and RDC-Integrated) are described. Section 5 presents the simulation model and evaluation methodology considered, while Section 6 discuss the results. In Section 7, the conclusions are drawn and possible future work is described.

## 2. Related Work

Routing is a fundamental activity in WSNs. Since sensor nodes individually do not have enough power to send data directly to the sink node, they need to collaborate by forming a multi-hop routing infrastructure, which should be as efficient as possible. This issue has been extensively studied and solutions such as Directed Diffusion [9], LEACH [8], and SPIN [10] deal with this problem. Nevertheless, the routing tree, which was evaluated in many studies such as EF-Tree [6], One-Phase Pull Diffusion [7], and SAR [15] is the most simple and widely used infrastructure. This tree is normally created and maintained by a decision process coming from the sink node, in a proactive fashion.

Another fundamental function in WSNs is density control. When battery recharge or replacement is not possible or hard to be performed, dense networks are formed to have more energy available and be more fault tolerant. In such configuration, many sensor nodes work on the same sensing event, generating redundant data. This is undesirable because it leads to more network traffic, probably causing more interferences, and increasing the latency of the network and the energy consumption. In this context, the role of the density control is to manage the network redundancy in an automatic manner, keeping only a minimum set of sensor nodes in activity at a certain time [18, 19]. As a result, the network lifetime is extended and the traffic is reduced. Regarding the existent approaches, although there are some proposals which centralize the calculations and decisions [1, 14, 13], most of them are distributed, in accordance with the WSN nature. Examples of distributed approaches are ASCENT [2], PEAS [18], CCP [17], OGDC [19], and the the algorithm described in [16].

Recently, a new research area has been opened for improving the performance of wireless networks, generically termed “cross-layer design”. The idea is to promote performance gains by jointly designing network protocols which would traditionally reside in different layers. When together, the protocols can interact and exchange important information. For integrating MAC and network (routing) layers in WSNs, for example, there are already some cross-layer proposals. In [4], a unified framework called MINA is presented, which encompasses network organization in groups, MAC and

routing protocols. In [20], a contention-based MAC protocol is designed to be integrated with GeRaF, an on-demand geographical routing protocol. In [12], a solution for random scheduling nodes is supported by interactions from the MAC and routing protocols.

Even though there are already many efforts at cross-layer design, few researches have yet shown concern about the interaction between density control and routing, which is the focus of the present work. In ASCENT [2], the authors observe that packet loss may occur when a node is set from the active to inactive state. They suggest that an improvement could be made by informing the network layer about ASCENT’s state changes, so as to let the routing infrastructure be repaired. No solution has been derived in detail and evaluated, though. Another work which deals more deeply with the cooperation between density control and routing is Tian and Georganas’s [16] work. In this case, the proposed density control algorithm is evaluated in conjunction with LEACH [8], a cluster-based routing algorithm for WSNs. Their approach is to insert the density control phase just before the LEACH set-up phase, i.e., to synchronize both functions, so that inactive nodes do not participate in the cluster forming and in the communication infrastructure. In the present work, we consider the same idea of synchronization, but we go further, providing a more precise evaluation of this kind of solution and also discussing the practical aspects involved in this application. Besides, we propose a fully integrated solution as a second and better alternative.

### 3. Integrated Design

The main motivation for integrating density control and data routing lies in the impact that the first causes on the performance of the second when both activities are considered separately. An integrated design promotes the reduction of this effect, allowing significant improvement on the system performance.

One of the effects caused by density control is the increase in the network dynamics. Density control changes the activity pattern of the sensor nodes, which are often put to sleep. Besides interrupting their sensing tasks, nodes in sleep mode have their radios turned off and thus can not participate on routing. Therefore, at any moment, the density control function may disable a router node, damaging the routing infrastructure and, as a consequence, causing data losses while this infrastructure is not repaired. Taking into account that the density control process is often distributed and dependent on the interactions among the nodes, the time that it takes to converge to a solution is variable and

can not be predetermined. As a result, the configuration of the routing rebuilding period is a hard task.

Another effect of density control over routing is related to the modification in the network traffic. Even though density control reduces the number of nodes in activity and thus the traffic generated by the application and the routing protocol, it introduces its own traffic flow: control messages need to be exchanged in order to let nodes interact among themselves. These additional control messages may cause more drops, delays, and energy consumption, which can affect the functions themselves. For example, control packet losses can lead to an incomplete routing infrastructure or an inefficient density control.

Taking into account the different impacts of density control over routing, the integration of these activities is a logical approach. This integration, which can be understood as a cross-layer approach, allows a more efficient interaction of these activities, since they can be designed to work in a synchronized way: the routing infrastructure is updated whenever the density control changes the topology. Also, they can share information and messages, reducing the network traffic.

## 4. Integrated Solutions

This section describes two solutions we designed to integrate routing and density control. They present a motivation for applying the cross-layer design and illustrate some aspects involved on the use of this kind of approach. In particular, they combine a density control algorithm called OGDC [19] — an approach that minimizes energy consumption without coverage degradation — with a classic solution of tree routing for WSNs. For the tree routing, we have based our implementation in the EF-Tree algorithm, which is described in [6].

The first evaluated approach, called **Synchronized Routing and Density Control**, or simply **RDC-Sync**, is an integration alternative for reducing the topology dynamics problem with an independent operation of density control and routing. The second one, called **Integrated Routing and Density Control** protocol, or simply **RDC-Integrated**, is a fully integrated solution, in which the routing and density control protocols are combined in a single one.

In the following, a description of the routing and density control algorithms considered is provided. Next, the strategies for integrating these algorithms are discussed.

## 4.1. Routing Algorithm: EF-Tree

EF-Tree works as follows. The sink node starts the tree building process by broadcasting a control message to its neighbors. When a node receives this control message for the first time, it identifies the message source as its parent and also broadcasts a control message to its neighbors. Whenever a node has data to transmit (locally generated or to be forwarded), it sends it directly to its parent. This building process is periodically repeated and its periodicity depends on how frequent the topological changes are. More dynamic networks need shorter rebuilding periods.

Note that the source of the first control message is chosen as the node’s parent in the tree, according to the original EF-Tree description. But it is possible to define other possibilities to choose a parent node as well. Depending on the application requirements, different rules can be applied for electing the best candidate, such as the closest node, the node with the highest energy stock, the node that belongs to a path with the highest amount of energy available, etc.

## 4.2. Density Control Algorithm: OGDC

The Optimal Geographical Density Control algorithm (OGDC) is a distributed density control solution. The idea is to make the sensor nodes temporarily inactive (i.e., in sleep mode) when they are not essential for guaranteeing coverage and connectivity of the network.

The OGDC algorithm is a localized algorithm in the sense that each node uses only local information to carry out the density control process and to decide whether to participate in the network coverage (be active) or not (sleep). The algorithm works in rounds; at each round the density control process is repeated and the set of active nodes is recomputed. The decision of each node is based on the position of other neighboring nodes that have already decided to be active in that round. They become active only if they contribute with maximum additional coverage to the existing set of active nodes.

During the operation of the algorithm, the nodes can be in three states: UNDECIDED, OFF and ON. In the UNDECIDED state, a node has not yet decided whether to stay active or not in the present round. If a node goes to the ON state, it stays active until the next round; if it goes to the OFF state, it stays inactive, turning off its devices in order to save energy. In a round, the global OGDC process finishes when all nodes have entered the ON or OFF states.

In a round, the density control process works for

each node according to Algorithm 1<sup>1</sup>. In the main part (first block), the nodes probabilistically decide to volunteer as a “starting node”. Starting nodes are responsible for starting the density control process by propagating POWER-ON messages.  $p_0$ ,  $T_D$ , and  $T_S$  are constants; they need to be carefully configured for the volunteering process to be efficient. In the second block, it is shown the procedure performed when a POWER-ON message arrives. The random time  $rt$  is very important to the optimality of OGDC<sup>2</sup>. It is computed using a formula which ensures that the node whose presence is most beneficial to the network at that time — the one which is closest to the optimal — gets the lowest valued back-off timer. Finally, the third block presents the action performed when a message drop occurs, which is the condition used by OGDC to assume that the network is excessively dense.

---

### Algorithm 1: OGDC pseudo-code

---

```

/* Main body, executed at the start of each round in node
i */
begin
  set  $state_i$  to UNDECIDED;
  set  $p_i$  to  $p_0$ ;
  while  $state_i$  is UNDECIDED do
    decide to be a starting node with probability  $p_i$ ;
    if node  $i$  decided to be a starting node then
      wait for a random time in  $[0, T_D]$  or until a
      POWER-ON message is received;
      if no POWER-ON message received then
        broadcast a POWER-ON message to all
        neighbors;
        set  $state_i$  to ON;
      else
        wait for a random time in  $[0, T_S]$ ;
        double  $p_i$ ;
    end
  end
end

/* Executed when node  $i$  receives a POWER-ON message */
begin
  add message source node to neighbors’ list;
  if the coverage area of node  $i$  is completely covered by
  the nodes in the neighbors’ list then set  $state_i$  to OFF;
  else
    wait for a random time  $rt$  or until other POWER-ON
    message is received;
    if no other POWER-ON message is received then
      broadcast a POWER-ON message to all neighbors;
      set  $state_i$  to ON;
    end
  end
end

/* Executed when a POWER-ON message is dropped in node  $i$ 
*/
begin
  set  $state_i$  to OFF;
end

```

---

<sup>1</sup>This pseudo-code was created by the authors of the present work based on the original description of OGDC.

<sup>2</sup>This variable name  $rt$  was created here to simplify the description. It is equivalent to a combination of many variables presented in the OGDC paper.



### 4.3. Synchronization Approach: RDC-Sync

A first alternative for integrating density control (OGDC) and routing (EF-Tree) processes is to implement them independently, but configure them to act in a synchronized way. This synchronization is made to renew the routing infrastructure whenever the density control algorithm changes the set of active nodes. So, in this approach, the time periods of these protocols have to be configured with the same value. Furthermore, in each time period, the routing tree construction has to be programmed to start immediately after the density control process is done.

The problem with this kind of approach is that the time synchronization is hard to be obtained in practice. Thinking of a density control process that is distributed and depends on interactions among the nodes, the time spent to converge to a solution is variable and can not be determined without a global view of the network. This implies the introduction of extra mechanisms to provide this synchronization support, which is complex and expensive for the network. For this reason, if the RDC-Sync is to be implemented in a real WSN, possibly the network designer will choose to estimate a global synchronization point. As a result, the performance of a real synchronization solution will depend on the estimate quality of the synchronization point. If OGDC finishes after this point, part of the tree construction will happen concurrently with OGDC, risking its quality. On the other hand, if OGDC finishes before the synchronization point, during the time interval between the finish time of OGDC and the synchronization point, information may be lost because the tree can be invalid.

The estimate quality of the global synchronization point can be measured by considering the best and worst possible cases for it. A real synchronization solution will have performance lying between these cases. The best possible synchronization situation, i.e., the one that gives a superior performance limit, is that in which the synchronization point is set to the time when the density control process finishes in the whole network. It can only be obtained if a global view of the network is available. The worst possible synchronization case, on the other hand, results in the inferior performance limit. The worst situation happens when the synchronization point is set to the beginning of the round, when the density control and routing tree process start simultaneously. In this case, the routing infrastructure is rebuilt based on a changing topology, which means that the integration level is zero. We refer to the best case as **RDC-Sync-B (Synchronized Routing and Density Control - Best Case)**, and

to the worst case as **RDC-Sync-W (Synchronized Routing and Density Control - Worst Case)**.

### 4.4. Fully Integrated Approach: RDC-Integrated

The problem with a synchronization approach like RDC-Sync is that the global synchronization point has to be fixed in a pre-defined value, which is only an estimate of the time when the density control process is supposed to finish. As a result, performance losses will still happen since the impact of the topology dynamics will not be completely avoided. Furthermore, even though a perfect synchronization could be obtained in practice, the sensor node would need the sum of the routing and density control time durations to finish these activities. The higher this time, the less efficient the network is, because important information collected by the sensor nodes can be lost during this time interval. Therefore, the ideal situation is the one in which density control and routing process perform concurrently. Another drawback of the RDC-Sync approach is that the synchronization point is global, meaning that the sensor nodes are given a new parent in the tree only if the density control process is finished on the whole network. As a result, even if a node has already changed to an ON or OFF state, its parent will not be set until all the nodes have already changed to one of these states. This is another reason for information loss and performance degradation when applying the synchronization approach. Last but not least, synchronization approaches as RDC-Sync do not share messages between the protocols and thus do not reduce the traffic.

A better integration alternative is the one that combines density control and routing in the same process, rather than putting one protocol on top of the other. This is the main idea of RDC-Integrated, which integrates EF-Tree routing and OGDC in a single solution. The result is a protocol that builds the routing tree using only the original OGDC messages, without incurring additional costs. Unlike RDC-Sync, RDC-Integrated is not based on a global synchronization point and a node's parent can be updated in the middle of OGDC process, at the time when the node receives the first control message.

To include the tree construction mechanism inside OGDC, some modifications to this algorithm are needed. First, the density control process has to be initiated by the sink, i.e., the sink has to be defined as the only starting node, rather than having the starting nodes chosen by a probabilistic volunteering process. As a consequence, in RDC-Integrated the sink

will be the first node to send a POWER-ON message and thus the following POWER-ON messages sent will flow from its position to the network boundaries. The main idea behind RDC-Integrated is that routing trees generally start from the sink, as in EF-Tree. By following this flow pattern, the POWER-ON messages can be used not only for density control but also for the tree construction. The POWER-ON messages in OGDC are originated by nodes just entered the ON state, and which will hence stay active in the round. For this reason, upon receiving a POWER-ON message, a node can include the message source as a candidate for being its parent in the tree. When data need to be routed, the parent is used as a path to reach the sink. Except for the tree construction mechanism, all the other functions of EF-Tree are kept unmodified inside RDC-Integrated.

---

**Algorithm 2:** RDC-Integrated pseudo-code

---

```

/* Main body, executed at the start of each round in node
i */
begin
  set statei to UNDECIDED;
  set parenti to nil;
  if node i is the sink then
    broadcast a POWER-ON message to all neighbors;
  else
    while statei is UNDECIDED do wait for POWER-ON
    messages;
  end
end
/* Executed when node i receives a POWER-ON message */
begin
  if parenti is nil then set parenti to the message source
  node;
  add message source node to neighbors list;
  continues as in the original OGDC algorithm;
end

```

---

Algorithm 2 shows the modifications made to OGDC with the purpose of including the tree construction mechanism. To simplify, the unmodified code of EF-Tree and OGDC was omitted.

## 5. Simulation Model and Evaluation Methodology

Simulations were conducted to evaluate the performance of the integrated solutions and show practical results in the use of integrated design. We used ns-2 (Network Simulator 2) [11], the widest adopted simulator for wireless networks.

### 5.1. Simulation Model

The simulation parameters were chosen based on the hardware of Mica2 nodes [3], which are commercially available. Table 1 shows the radio parameters and their respective values, as used in the simulations. Mica2 has the same radio module for sensor nodes and sink

Parameter	Value
Transmission Power	45.0mW
Reception Power	24.0mW
Idle Power	24.0mW
Bandwidth	19,200 bps
Communication Range	40m

**Table 1. Simulation parameters based on Mica2 nodes.**

nodes, so we used the same parameters for both types of nodes.

Regarding that the Mica2 nodes can work with a temperature sensor, the application chosen to be simulated was the temperature monitoring of a square area. In this application, the Mica2 nodes sense continuously and report periodically their results to the sink node, which is located at the center of the area. The periodicity of report was set to 20 s. All the nodes begin their sensing activities in random times between 0 and 20 s. The transmitted data packets have 32 bytes, as in Mica2 operating system (TinyOS). The sensing range considered was 20 m, which is exactly half of the communication range. The MAC layer was the IEEE 802.11 version available in ns-2, as the Mica2 nodes implement a CSMA/CA protocol. As it is possible to connect a sensor to the Mica2 sink, we assume that our sink node is also capable of sensing.

For EF-Tree and OGDC, the control packets have 32 bytes. The OGDC protocol was configured with a round time of 100 s. Regarding its constants and timeout values, they are the same as the ones used in [19].

Concerning the network setting, the nodes are static and were randomly distributed in the area, according to a uniform distribution. The initial energy of each node was set to 100 J. This value is sufficient for the network to survive without node loss for more than 3,000 s, in all simulated settings. Number of nodes and network density were varied in the simulation experiment, so their values will be described later.

### 5.2. Evaluation Methodology

For the purpose of evaluating the benefits of integrating density control and routing, four solutions are considered, as described in the following:

**Routing Tree (RT).** Implements the tree routing algorithm (EF-Tree) without density control, with the routing tree updated every 100 s.

**Synchronized Routing and Density Control (RDC-Sync).** Implements a synchronization approach of density control and routing algorithms, as described in Section 4.3. In order to evaluate the re-

sults of this approach, the best (RDC-Sync-B) and worst (RDC-Sync-W) cases are the scenarios considered, which give the superior and inferior performance limits for a real synchronization point, respectively. In their implementation, we consider that the sink node does not participate in the OGDC process (because it can not be turned off), but contributes to the network coverage by keeping its sensor permanently turned on.

**Integrated Routing and Density Control (RDC-Integrated).** Implements a fully integrated approach of density control and routing algorithms, as described in Section 4.4.

For the purpose of measuring and comparing the performance among the adopted solutions, some metrics were elected, as defined in the following:

**Packet delivery ratio.** Relation between the number of data packets received by the sink and the number of packets sent by the sensor nodes. This metric indicates the amount of information from the environment that is correctly delivered to the network observer.

**Total consumed energy.** Total energy consumed by all sensor nodes throughout the simulation time period, not including the sink node. It indicates the cost in energy of each solution.

**Delay.** Average time that all data packets spend between their transmission by the sensor nodes and their reception by the sink. If the delay is high, this means that the information delivered to the network observer is late and may not be useful anymore.

**Coverage (global view).** Percentage of the area that is sensed by at least one sensor node. Since it varies in time, the value of this metric is the average of the values collected every 20 s of simulation, which is the data generation time period.

**Coverage (sink view).** Similar to the previous metric, but in this case the coverage is measured in the sink, i.e., considering only the information effectively received by the sink. As a result, this metric is the real network coverage, being a function of the global coverage and the data delivery performance of the network. It is also collected every 20 s.

**Nodes in activity.** Average number of nodes which have decided to be active in a round. When comparing the solutions, if the global coverage is the same, a high number of active nodes may be an indication that the network redundancy is above the minimum needed.

## 6. Evaluation Results

To evaluate the performance of the integrated solutions, two sets of simulation experiments were con-

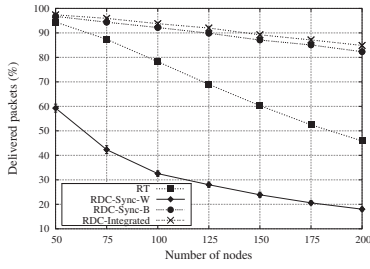
ducted. The purpose of the first set was to provide a performance comparison among these solutions when the network size varies and all the nodes are alive (i.e., without faults). The second set, in turn, was designed to allow a comparative analysis of the network coverage along the simulation time, also offering results in the network lifetime obtained for each solution.

### 6.1. First Set: Performance Evaluation Without Faults

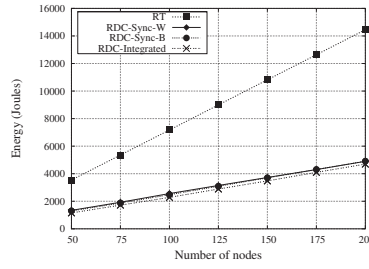
In order to compare the performance of the chosen solutions, simulations were conducted varying the number of nodes from 50 to 200. The area dimension was varied simultaneously in order to keep a fixed density of 5 nodes per 1,000 m<sup>2</sup>. In this set, all the simulation experiments were executed for 3,000 s and repeated 33 times. The results present confidence intervals of 95%.

Fig. 1(a) shows the packet delivery ratio per network size, for all solutions. We can see a better performance of RDC-Integrated, which is the only solution that completely avoids the invalidation of routes due to the dynamics introduced by density control (when a node state is switched from active to inactive). In RDC-Sync-B, this effect of density control is only partially avoided because packets may be lost before the global synchronization is reached, i.e., before the density control process has finished in all nodes. In RDC-Integrated, the routes are computed inside the density control process, which reduces the time interval when routes may be invalid. Nevertheless, although the difference among these solutions is small, the global view in RDC-Sync-B is hard to be achieved. Thus, we can expect real performance to lie between RDC-Sync-B and RDC-Sync-W results, depending on the estimate quality of the synchronization point. In RT, we can see a delivery ratio near that of RDC-Integrated for smaller network sizes. However, as the number of nodes is increased, the network is negatively impacted by the high amount of redundant data generated.

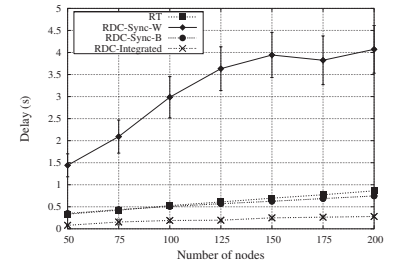
Fig. 1(b) shows the energy consumption results. Obviously, in the solutions that apply density control, less energy is spent due to the lower number of active nodes. Moreover, these solutions reduce redundant data and traffic consequently, saving even more energy. If the density control is fully integrated with the routing construction, the energy efficiency is even better, as the results for RDC-Integrated in Fig. 1(b) show. The application of this strategy reduces the number of control messages by integrating the messages of density control and routing tree construction, which are tasks performed independently in the other approaches. In addition, as we considered the effective participation of the



(a) Packet delivery ratio.



(b) Total consumed energy.



(c) Delay.

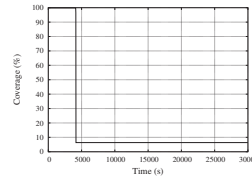
**Figure 1. Performance comparison of distinct integration solutions per number of nodes.**

sink in the density control process of RDC-Integrated, the number of active nodes is lower (see Fig. 2(c)), and this improves the energy savings.

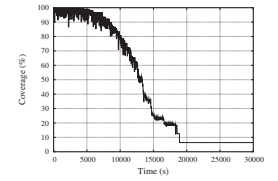
In Fig. 1(c), we can see the delay resulting from the application of each solution. RDC-Sync-W has the worst performance due to the high presence of invalid routes. Data packets are retained in the routing queues for retrials, and this causes the increase on the delay of successfully received packets, which are few in this case. RDC-Integrated has the best performance because in this solution, as soon as the topology is altered the routing tree is rebuilt and data can be correctly and immediately forwarded.

Regarding the resulting area coverage, we can see in Fig. 2(a) that the better global coverage is obtained with RT, which does not perform density control, and so all nodes remain active (see Fig. 2(c)). The other solutions also present a high area coverage, with a difference of 3%, but a lower performance is observed in RDC-Sync-W. This is explained by the impact of the routing tree construction that is performed concurrently with the density control process, causing more collisions and control packet losses. As described in Section 4.2, in OGDC these collisions can make some nodes become inactive, because it is believed that if the number of collisions is high, it means that the network is dense. Nevertheless, collisions may have other causes and, as a result, OGDC could negatively affect the global coverage.

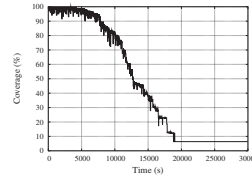
If coverage is measured in the sink (see Fig. 2(b)), we can also see a good performance in all solutions but RDC-Sync-W, due to its low delivery ratio. Note that although the use of RT implies in a low delivery ratio, the coverage is maintained as a result of data redundancy. Nevertheless, when the network size is increased, the drawbacks of this strategy become clearer. With less than 50% nodes (see Fig. 2(c)), the other solutions manage to obtain almost the same coverage



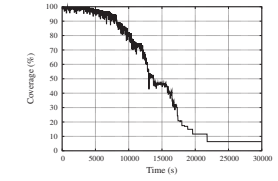
(a) RT



(b) RDC-Sync-W



(c) RDC-Sync-B



(d) RDC-Integrated

**Figure 3. Coverage (global view) in time.**

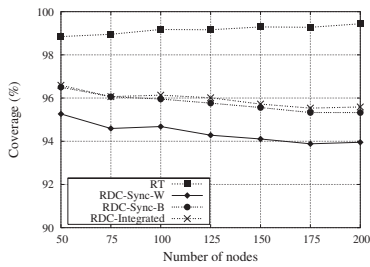
with reduced network traffic and energy consumption.

## 6.2. Second Set: Performance Evaluation With Faults

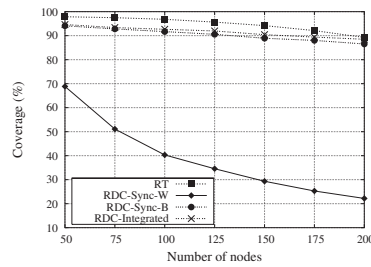
In order to verify the impact of each approach on the network coverage in time and network lifetime, simulations were conducted without limiting the simulation time, i.e., they were executed until all the sensor nodes have run out-of-energy. For these experiments, the number of nodes was fixed in 100, with 5 nodes per 1,000 m<sup>2</sup>. The results present the coverage in time for one simulation of each scenario.

Fig. 3 shows the coverage along the simulation time considering a global view of the network. We can see the great benefit of density control on the network life-

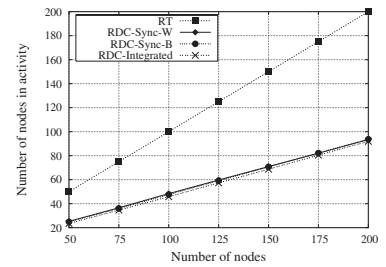




(a) Coverage (global view).

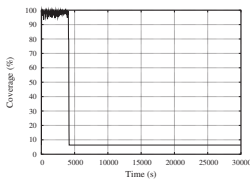


(b) Coverage (sink view).

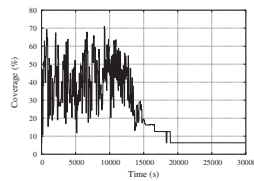


(c) Nodes in activity.

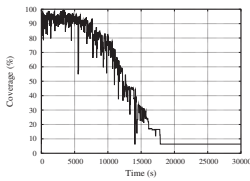
**Figure 2. Coverage and activity per number of nodes.**



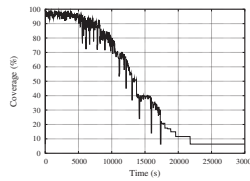
(a) RT



(b) RDC-Sync-W



(c) RDC-Sync-B



(d) RDC-Integrated

**Figure 4. Coverage (sink view) in time.**

time. While with RT the network loses its coverage completely before 5,000 s (discarding the sink coverage), the other solutions manage to increase the network lifetime in four times. The behavior of the solutions with density control, regarding area coverage, is mainly a consequence of the number of alive nodes (i.e., nodes which still have energy) maintained by the solutions. Nevertheless, RDC-Integrated presents a better performance, keeping the highest coverage most of the time. This lifetime extension occurs due to its energy savings, as shown before.

When we observe the sink coverage, as shown in Fig. 4, both the differences among the integration strategies and the performance of the routing protocol become clearer. As we can notice among the solutions with density control, RDC-Sync-W shows the worst

performance, as expected. The delivery ratio makes the difference: RDC-Sync-W shows low sink coverage with high variance; and RDC-Integrated presents lower variance when compared to RDC-Sync-B. Here we can also see a benefit of the low energy savings of RDC-Integrated because the network lifetime was extended.

## 7. Conclusions and Future Work

Wireless Sensor Networks (WSNs) differ from traditional networks in the sense that they have very limited resources. This poses challenges to the elaboration of solutions for these networks because they need to have not only a correct but also an efficient operation. Furthermore, some extra network functions need to be introduced in order to provide energy savings such as density control, which causes special dynamics that is hard to be managed and affects the performance of data delivery in the network.

This paper contributes to the development of WSNs in the following aspects. First of all, we analyze the impact of density control over routing, where it becomes clear that it is important to consider the aspects of both functions as integrated. Secondly, we propose two distributed design approaches and apply them in specific solutions (RDC-Sync and RDC-Integrated) to provide a better network operation. Last but not least, through simulation results, we offer a comparative analysis that demonstrates the benefits of this kind of integration design for satisfying the network requirements.

In particular, the RDC-Integrated solution shows the benefits of a more ambitious integration design, arranging routing and density control as a single network function, since the simple strategy of synchronization is hard to be obtained in practice, and, hence, insufficient. Although the main purpose of this paper is to motivate the integrated design, the RDC-Integrated solution can be regarded as a contribution itself, since it has some

properties with interesting practical use. First of all, it demands little implementation effort. Furthermore, it can be easily extended not only to consider other metrics for constructing the tree infrastructure, but also to add even more functions to OGDC control messages, such as network management and other self-organizing processes.

Besides extending the RDC-Integrated solution, there are other open issues. A next step could be to synchronize the application timing with density control and routing, i.e., to schedule the transmission of data only after the density control and the construction of the routing infrastructure are done. Another possibility is to work with the integration of different density control and routing solutions.

## References

- [1] M. Cardei, D. Maccallum, X. Cheng, M. Min, X. Jia, D. Li, D. Zhu Du, and D. Du. Wireless sensor networks with energy efficient organization. *Journal of Interconnection Networks*, 3:213–229, 2002.
- [2] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1278–1287, New York, USA, 2002.
- [3] Crossbow. Mica2 - wireless measurement system, Access: February 2004. [Online] Available: <http://www.xbow.com/>.
- [4] J. Ding, K. Sivalingam, R. Kashyapa, and L. J. Chuan. A multi-layered architecture and protocols for large-scale wireless sensor networks. In *Proceedings of the 58th IEEE Semianual Vehicular Technology Conference*, volume 3, pages 1443–1447, 2003.
- [5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the 5th Annual International Conference on Mobile Computing and Networks*, pages 263–270, Seattle, Washington, USA, August 1999. ACM Press.
- [6] C. M. S. Figueiredo, E. F. Nakamura, and A. A. F. Loureiro. Multi: A hybrid adaptive dissemination protocol for wireless sensor networks. In *Proceedings of the 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks*, volume 3121 of *Lecture Notes in Computer Science*, pages 171–186, Turku, Finland, July 2004. Springer.
- [7] J. Heidemann, F. Silva, and D. Estrin. Matching data dissemination algorithms to application requirements. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pages 218–229, Los Angeles, CA, USA, November 2003. ACM Press.
- [8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the Hawaii International Conference on System Sciences*, pages 4–13, January 2000.
- [9] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *ACM/IEEE Transactions on Networking*, 11(1):2–16, February 2002.
- [10] J. Kulik, W. Heinzelman, and H. Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks*, 8(2/3):169–185, 2002.
- [11] NS-2. The network simulator - ns-2, Access: February 2004. [Online] Available: <http://www.isi.edu/nsnam/ns/>.
- [12] M. L. Sichitiu. Cross-layer scheduling for power efficiency in wireless sensor networks. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.
- [13] I. G. Siqueira, L. B. Ruiz, A. A. F. Loureiro, and J. M. S. Nogueira. Coverage area management for wireless sensor networks. In *International Journal of Network Management. To appear.*, 2006.
- [14] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications*, volume 2, pages 472–476, 2001.
- [15] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27, 2000.
- [16] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 32–41, Atlanta, GA, USA, 2002.
- [17] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems*, pages 28–39, Los Angeles, California, USA, 2003. ACM Press.
- [18] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, pages 28–37, Providence, Rhode Island, USA, May 2003.
- [19] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Wireless Ad Hoc and Sensor Networks: An International Journal*, 1(1–2):89–123, January 2005.
- [20] M. Zorzi. A new contention-based mac protocol for geographic forwarding in ad hoc and sensor networks. In *Proceedings of the IEEE International Conference on Communications*, volume 27, pages 3481–3485, 2004.