# Energy Aware Self-Organizing Density Management in Wireless Sensor Networks

Erwan Le Merrer, Vincent Gramoli, Anne-Marie Kermarrec, Aline Viana, Marin Bertier

# Energy Aware Self-organizing Density Management in Wireless sensor networks

Erwan Le Merrer
France Telecom and
IRISA, France
erwan.lemerrer@orange-ft.com

Vincent Gramoli,
Anne-Marie Kermarrec,
Aline C. Viana
IRISA/INRIA, France
Aline.Viana@irisa.fr

Marin Bertier
IRISA/INSA, France
Marin.Bertier@irisa.fr



## ABSTRACT

Energy consumption is the most important factor that determines sensor node lifetime. The optimization of wireless sensor network lifetime targets not only the reduction of energy consumption of a single sensor node but also the extension of the entire network lifetime. We propose a simple and adaptive energy-conserving topology management scheme, called SAND (Self-Organizing Active Node Density). SAND is fully decentralized and relies on a distributed probing approach and on the redundancy resolution of sensors for energy optimizations, while preserving the data forwarding and sensing capabilities of the network. We present the SAND's algorithm, its analysis of convergence, and simulation results. Simulation results show that, though slightly increasing path lengths from sensor to sink nodes, the proposed scheme improves significantly the network lifetime for different neighborhood densities degrees, while preserving both sensing and routing fidelity.

## Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*distributed networks, wireless communication*

## General Terms

Design, Management

## Keywords

Wireless sensor networks, topology management, peer-to-peer distributed systems.

## 1. INTRODUCTION

**Context.** Area monitoring is one of the most typical applications of wireless sensor networks (WSNs). It consists in deploying a large number of sensors in a given geographic area, for collecting data or monitoring events. It is not unusual that in this situation, human intervention is not feasible. Sensors are then thrown in mass, for example from a plane, and must be able to form a network and to operate in a decentralized self-organized manner, maintaining connectivity and area monitoring as long as possible. In addition, because of the absence of wire and the small physical size of sensors, WSNs have strong power restrictions. Mechanisms for energy optimization in WSNs constitute then an important requirement. This optimization targets not only the reduction of energy consumption of a single sensor node but also the extension of the entire network lifetime. The sensor network lifetime is defined as the period during which the routing fidelity and the sensing fidelity of the network are guaranteed. Guaranteeing sensing fidelity means that any monitored stimulus in the area will always be sensed by at least one sensor. Routing fidelity means the existence of a path between any sensor node and *at least* one base station. Our goal is then to leverage node redundancy in WSNs to reduce and distribute the computational and communication energy consumption of the network between sensors. We consider that the cooperative nature of sensors offers significant opportunities to manage energy consumption.

Given the potentially large number of sensors in a WSN and their limited resources, it is also crucial to deploy fully decentralized solutions and to evenly spread the load over the network. Recent works in Peer-to-Peer (P2P) systems can be successfully adapted to explore new solutions for energy consumption distribution in sensor networks.

**Contributions.** This paper describes an adaptive and fully decentralized topology management scheme, called SAND (Self-organizing Active Node Density). SAND significantly extends network lifetime by reducing nodes activity. The major contribution of SAND is its simple algorithm: each node takes local decision based on the observation of its neighborhood in order to ensure the properties of routing and sensing fidelity. Moreover, SAND as a whole converges towards the expected properties. In order to manage energy consumption, we leverage P2P's cooperation paradigms and explore local node information and sensors resolution through neighborhood communication only. No node location information is required, and our approach is independent of any wireless routing protocol.

For any configuration of the energy-aware topology, we show that SAND guarantees that routing and sensing fidelity will be extended. In comparison with the case where no topology management is applied, SAND considerably ex-


*MobiShare'06,* September 25, 2006, Los Angeles, California, USA.

tends network lifetime at the price of the slightly increasing paths lengths from sensor to sink nodes. Moreover, simulations of SAND suggest that network lifetime and SAND's robustness increases proportionally to node density.

**Outline.** The rest of this paper describes and evaluates SAND. In Section 2, we discuss the advantages in performing power management of radios in wireless networks and describe our system model. We review related works in Section 3. In Section 4, we describe the SAND approach, analyse interesting properties, and discuss design issues. We present the performance results in Section 5. Finally, in Section 6, we conclude this paper and discuss future works.

# 2. FOREWORD

## 2.1 Where does the energy go?

Power dissipation analyses of a sensor node in the literature show that wireless communication is a major energy consumer during system operation [1, 2]. Results show that (*i*) the overhearing process increases power consumption,[1] and (*ii*) energy optimizations must turn off the radio and not simply reduce packet transmission and reception. SAND, therefore, incorporates power management into the communication process. We optimize energy consumption by completely turning off radio whenever possible, conserving energy both in *Idle* state when no traffic exists and in overhearing due to data transfer.

We also take advantage of the sensor network density for energy savings. SAND also powers down sensor nodes that are equivalent from a sensing perspective. In summary, SAND is a very simple topology management scheme, which generates low communication overhead. SAND applies in the context of nodes that are able to:

1. turn off their radios for *communication power conservation*, while still maintaining connectivity between sensors and sinks, *i.e.* the routing fidelity;
2. completely power down for *sensing power conservation*, while still ensuring a correct stimuli monitoring, *i.e.* the sensing fidelity.

## 2.2 System model

We consider a distributed system consisting of a finite set of $n$ sensors and $s$ sinks, each uniquely identified. Nodes (sensors or sinks) are spread into a delimited area. We consider that both sensor and sink nodes form a connected network or that all network partitions contain at least one sink node. Nodes may crash and recover during the network lifetime, we do not consider Byzantine failure. We assume that, although there is no need for synchronized clocks, there is an upper bound on the drift rate of local clocks. We define $\delta$ as an upper bound on the transmission time of a message between two neighbors. We consider that sensing and communication ranges are equal.

SAND does not require any node location information. We explore density determination by assuming that nodes communicate only by 1-hop broadcast toward nodes in their neighborhood, corresponding to their transmission range. Finally, a node may be in one of the four following energy states: sleep, sensor-only, router-sensor, and gateway. Each state corresponds to a defined node activity and one or several services provided to the network (see Table 1). SAND guides nodes' energy state switching independently of the underlying wireless routing protocol. Interactions between SAND and the routing protocol are part of a future work.

[1]In [3], authors show that for typical sensor network scenarios, around 65% of all packets received by a sensor node need to be forwarded to other destinations.

# 3. RELATED WORKS

Topology management techniques, called SPAN [4] and GAF [5], have similar goals to those of SAND: they trade network density for energy savings while preserving the forwarding capacity of the network. Nevertheless, they do not exploit the absence of traffic in the active sensing state. Besides the energy consumption reduction in the sensing state, the STEM [6] scheme also proposes to ensure a satisfactory latency for transitioning to the routing state. Authors then suggest integrating STEM with scheme such as GAF and SPAN. SAND coordinates the radio sleep and the wakeup switch for sensing and/or routing states in a unique scheme, being no integration necessary.[2] Several algorithms have been proposed for exploiting the area coverage problem in sensor networks [7, 8, 9]. Contrary to SAND, all these solutions assume that the sensors are aware of their own positions. Authors in [8] do not address the connectivity problem and require every sensor to know all their neighbors positions before making its monitoring decisions [8]. The proposal presented by Carle and Simplot-Ryl [7] specifies that each sensor needs to construct its subset of relays and broadcast it to its neighbors, which generates higher communication overhead than SAND. The solution proposed by A. Gallais *et al.* [9] relies on low communication overhead and does not need any neighborhood discovery. Nevertheless, nodes have to memorize the positions and the decisions of their neighbors in order to make appropriate monitoring decisions. In SAND, however, nodes need a small amount of information (*e.g.* only a partial neighborhood discovery) and perform a low processing overhead to take their activity decisions. Clustering algorithms can be also used to select router nodes. As an example, M. Chatterjee *et al.* in [10] use quite sophisticated concepts and heuristics to decide which nodes should be cluster heads. SAND instead, has as a major contribution the simplicity for router election procedure.

# 4. SAND DESIGN RATIONALE

A major issue in sensor-based applications is the diffusion of the sensed data to a specific entity that can store and process it – the sink node. Mechanisms that ensure this diffusion are important components in WSNs and have been the subject of many researches in the literature [11]. In this context, the cooperation between energy conserving and information diffusion robustness is crucial.

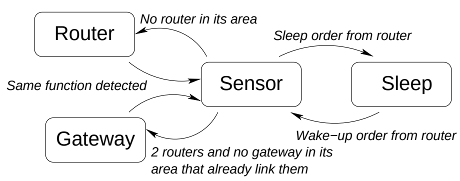


**Figure 1: Energy state transition diagram in SAND.**

[2]The insurance of a satisfactory latency for transitions of state is not our focus here.

**Table 1: Energy state description**

| State | Energy consumption | Activity | Provided service |
|---|---|---|---|
| *Sleep* | very low | none | - periodically turns on the radio for receiving control msg |
| *Sensor-only* | low | sensing | - sensing<br>- periodically turns on the radio for receiving/sending control msg<br>- sending local data, when needed |
| *Gateway* | high | sensing<br>routing | - sensing<br>- receiving/sending control msg<br>- sending/forwarding data |
| *Router-sensor* | high | sensing<br>routing | - sensing<br>- receiving/sending control msg<br>- sending/forwarding data<br>- managing sensor density |

We take advantage of information provided by routing layer and/or information related to the envisaged application to determine when the radio is not needed. We consider that during network lifetime, sensor nodes can alternate their energy consumption between four states (see Fig. 1): (1) *sleep*, where all hardware components are powered off, (2) *sensor-only*, where only sensor and some pre-processing circuity are powered on, and (3) *gateway* and *router-sensor*, where all hardware components are powered on.[3] SAND performs then the energy-aware topology management by controlling the routing and the sensing fidelity during the network lifetime.

## 4.1 Forwarding nodes distribution

The forwarding nodes distribution is performed in two consecutive phases. The first one distributes nodes in router-sensor state uniformly in the network. The second one consists in connecting close router-sensors by selecting nodes to switch to the gateway state. These two phases are based on the SONDe's principle [12]: if a node does not detect any neighbor in each one of these two states then it turns itself into the missing state.

**Algorithm 1: *The SAND Algorithm***

1: **Phase 1:**
2:   **if** $\neg$router-detection() **then**
3:     $status \leftarrow$ router-sensor
4:   **else if** $r \leftarrow$ router-detection() **then**
5:     **if** $r.ts > ts$ **then**
6:       $status \leftarrow$ sensor-only

7: **Phase 2:**
8:   **if** $|routers \leftarrow$ detected-routers()$| \geq 2$ **then**
9:     **if** $\neg(g \leftarrow$ gateway-detection()$) \vee (g.ts < ts)$
10:         $\wedge\neg(routers \subseteq g.routers)$ **then**
11:       $status \leftarrow$ gateway
12:     **else** $status \leftarrow$ sensor-only
13:   **else if** $status =$ gateway **then**
14:     $status \leftarrow$ sensor-only

**Algorithm overview.** The SAND algorithm is presented in Algorithm 1. Each period of time $\Delta > 2\delta$, the router-sensor nodes send `Hello` messages containing their current state and a timestamp $ts$. Observe that we do not focus on specifying the message retransmission in case of collision; we rather assume that this is implemented at a lower layer. The timestamp $ts$ of node $i$ contains: the time $i$ spent in its current state and its identifier (tie breaker). Each sensor-only node in the network checks if there is a router-sensor node in its immediate neighborhood, by listening during a timeout $T_{on}$ with $T_{on} > \Delta + \delta$. If no router-sensor node is detected, then the sensor-only node becomes a router-sensor node (Lines 2-3). If a router-sensor node detects the presence of another router-sensor node in its transmission range with a higher timestamp than its own, it comes back to the sensor-only state (Lines 4-6). While the first phase guarantees a good distribution of router-sensor, the second phase (Lines 8-14) elects gateways to connect them. A sensor-only node becomes gateway if it detects two routers and no gateway with a lower timestamp than its own and that already makes a link path between those two routers. A gateway node informs with a period $\Delta$ about the routers it sees. If a gateway detects another gateway which connects the same router-sensor and with a higher timestamp than its own, it comes back to the sensor-only state (Line 12).

[3] Router-sensor and gateway nodes can also optimize local energy consumption by changing the power state of their memory and/or processor.

**Independent-dominating set convergence.** In the following, we show that SAND presents two interesting properties borrowed from graph theory. First, to help routing to a sink node, each node is in the neighborhood of a router-sensor node or is itself a router-sensor node. Second, to prevent energy consumption waste, a subset of nodes becomes router-sensor nodes. There are solutions for related problems known as *vertex cover* and *minimal dominating set*, which guarantees activated sensors to form such a set.

The minimality problem of the aforementioned solutions might involve many state changes each time a router-sensor crashes. Here, we rather ensure that the sensor-router nodes satisfy both domination and independence properties. This helps at reducing, yet making it sufficient, the number of router-sensor nodes, while this number has not to be minimal. Roughly speaking, the router-sensor nodes satisfy *(i) dominance*: all nodes are either router-sensor or a neighbor of a router-sensor and *(ii) independence*: no router-sensor node is a neighbor of another router-sensor node.

The following proof shows that the algorithm converges to a configuration verifying both properties under system stabilization. Let a *real-router* (resp. *real-sensor*) denote a router-sensor node (resp. sensor-only node) that became router-sensor (resp. sensor-only) $T_{on}$ time ago, and which did not revert its state since then.

THEOREM 1. *The SAND algorithm converges towards an independent-dominating set.*

PROOF SKETCH. The proof is divided in three parts. First we show that the independent-dominating property is an invariant. Consider the communication subgraph containing only real-routers and real-sensors whose real-routers form an independent dominating set when the system stabilizes. That is, any real-sensor node receives a message from a real-router in each $T_{on}$ period of time and does not become a router. Similarly the real-routers stay in their state.

Second we show that independence can never be violated. Observe that nodes are initially in their sensor-only state and thus can not violate independence by awakening. Before stabilization, real-router can crash but independence is never violated since message delay remains bounded.

Finally, we show that the number of locations where the dominance is violated eventually decreases, let $\mathcal{G}$ be any communication subgraph whose set of real-routers is not dominant. After some time, some nodes of $\mathcal{G}$ become routers. This might happen in the meantime at different places in the same neighborhood. After a $T_{on}$ delay, messages are exchanged between routers and the one with the lowest $ts$ is chosen to become exclusively the real-router of the neighborhood. Other router nodes, so as sensor-only nodes, become real-sensor. □

Experiment described in Fig 2(a) confirms our theoretical analysis and shows the result of our router selection after three simulation rounds: this simulation leads to 50 router-sensor nodes (black circles) for 450 sensor-only nodes (gray circles). Provided these properties, we claim that a sufficient sensor nodes density will provide enough gateway candidates to ensure the connection between two close router-sensors. Consequently, SAND does not determine the optimal minimum number of forwarding nodes to maintain sink connectivities, ensuring then, that there are several paths between any node and at least one sink. This redundancy makes the routing fidelity more resilient to failures. Our claim is satisfied by the experiments obtained in Fig. 2(b) commented hereafter. Specifying a protocol to obtain a path among all router-sensors from a dominating set is left as an open work.

## 4.2 Sensing guarantee

Fidelity in stimuli sensing can be ensured only by router-sensor and gateway nodes, because they are uniformly distributed in the network and their sensing range is equal to their transmission range. Nevertheless, for reliability issues and for the cases where a specific sensor node density should be ensured, SAND allows the control of the sensor-only nodes resolution in each target area, while performing the sensing load distribution among nodes.

To this end, router-sensor nodes are in charge of selecting nodes to switch to sleep or sensor-only state. This selection depends on the envisaged reliability degree of the monitored area. Thus, nodes that are for a long time in the sensor-only state will be selected to switch to sleep state, and vice versa. Each sensor-only node periodically turns on its radio and sends `Hello` messages containing its current state and its estimated lifetime ($el$).[4] Nodes in sleep state also periodically turn on their radio, but never send messages.[5]

Upon reception of sensor-only $el$ of sensor-only neighbors, router-sensor nodes compute the average (by considering the $el$ of the last switched-to-sleep nodes too) and the standard deviation (by considering only the sensor-only nodes that have lower $el$ than the resulting average). Finally, router-sensor nodes send 1-hop "`switch-to-sleep`" order messages to sensor-only nodes that have their $el$ level lower than the resulting standard deviation. The switch of sensor-only to sleep state is performed as soon as another sensor-only node

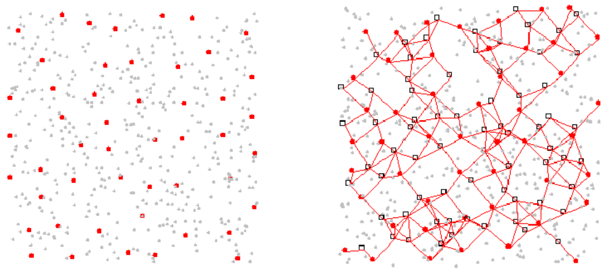

**Figure 2: A 300×300 $m$ area network with an average of 20 neighbors per node representing: (a) the router selection, (b) the energy state distribution.**

appears in the monitored area. Sensor-only nodes also turn on their radio if any local collected data has to be transmitted to the sink (*e.g.*, for full memory resource).

Router-sensor nodes also control the switch from sleep to sensor-only states, by sending "`switch-to-sensor`" order messages.[6] In this case, nodes in the sleep state switch to the sensor-only state if they have (1) their radio turned on, and (2) their $el$ level is higher than the computed average. In addition, like sensor-only nodes, nodes in the sleep state with the radio turned on, verify their router-sensor node connectivity. If no `Hello` message from a router-sensor node is received, a sleep node becomes a sensor-only node and then, based on the SAND bootstrap procedures, can switch to the router-sensor or gateway state. Fig. 2(b) shows the result of the SAND energy state distribution at a random point in time. The figure presents 68 router-sensor nodes (black circles), 47 gateway nodes (squares) and 385 sensor-only and sleep nodes (gray circles). The line connecting points represents the connectivity among forwarding nodes. The network connection among forwarding, sleep, and sensor-only nodes is not represented here. This shows that the generated forwarding topology is connected.

## 4.3 Discussion

We now discuss some design choices.

**Outlining parameters:** The radio of sleep and sensor-only nodes is periodically turned on for fidelity verifications at intervals $T_{off}$ and stay on for at least a timeout value $T_{on}$. The range of $T_{off}$ can be influenced by the time that nodes have been conserving their energy during the sleep state. Optimizations of these parameters are under evaluation.

**Outlining advantages:** Contrary to some existing area coverage solutions, SAND does not require nodes position information or geographic coverage computing for ensuring connectivity. Therefore SAND has a low computing overhead. Moreover, since nodes do not need to perform a complete neighborhood discovery to take their state decisions, SAND has also as advantage a low communication overhead.

**What if disconnections occur:** Considering the poor failure resilience of sensor nodes and the SAND's guarantee of sink connectivity, it may occur that failed nodes cause temporary disconnections. This can also occurs if `Hello` messages from router-sensor nodes are lost, causing gateway nodes to become sensor-only nodes. Nevertheless, since SAND allows disconnection detection and restores connectivity by state switching, we consider that disconnection periods will not be long. Nodes may store their messages and

[4] $el$ corresponds to the expected remaining node energy and is set by assuming that nodes will have a start energy level and they will consume energy (related to their state) until they die.

[5] We assume their estimated lifetime is the same of the last $el$ sent when they were in the sensor-only state.

[6] In the case of the reception of two contrary order messages, the priority is given to the "`switch-to-sensor`" messages.

wait for connectivity restoration. In the case of router-sensor `Hello` message losses, gateway nodes can be instrumented to wait at least a timeout value of $2 \times T_{on}$ before deciding to switch to sensor-only state. This kind of improvements is subject to future works.

# 5. PERFORMANCE EVALUATION

We have conducted a number of simulation experiments over 800 rounds, using a simulator consisting of the SAND engine and a network emulation environment. We experiment on a large scale static network with 4,700 nodes. Nodes are uniformly distributed over a square area of 700 $m$ on a side and have a transmission range of 37 $m$. The dynamism of the created topology is imposed by fail nodes only.

The simulator is a discrete time-based engine, in which actions are performed per round of simulation. We have set $T_{off}$ to be 2 rounds. We consider that nodes send order or `Hello` messages at each round. We also consider that sensor-only nodes send `Hello` messages at each $\Delta$ period, where $\Delta$ and $T_{on}$ are initially set to 1 round.

We set all nodes with an estimated lifetime $el$ of 100,000 unities ($u$). Our energy consumption model is based on the power consumption of the sensor node described in [13]. We consider thus, the costs described in Table 2. The network lifetime column shows approximated values for the estimate maximum lifetime corresponding to nodes in each state.

We uniformly generated 1,000 stimuli over the network. Unless otherwise specified, we set the reliability degree of each monitored stimulus to 5 nodes. No wireless routing protocol is implemented. Thus, to evaluate the routing fidelity of our approach, we simply verify if there is a path between a source node (*i.e.*, a sensor-only, a gateway, or a router-sensor node) and at least one sink node. Path verification is performed each time a stimulus is generated. Unless otherwise specified, sink nodes are uniformly distributed over the network and correspond to 1% of all nodes. To examine the impact of the neighborhood density in the network lifetime, we vary the number of nodes from 1,000, to 1,900, 2,800, and 4,700 (corresponding to approximately 10, 20, 30, and 50 neighbors in range, respectively), while keeping constant the area and the transmission range of nodes.

**Table 2: Energy consumption**

| Node state | Radio state | Energy consumption | Estimated lifetime |
|---|---|---|---|
| *Sleep* | radio OFF<br>radio ON | $\pm$ 10$u$<br>$\pm$ 70$u$ | ≅3300<br>*rounds* |
| *Sensor-only* | radio OFF<br>radio ON | $\pm$ 200$u$<br>$\pm$ 270$u$ | ≅450<br>*rounds* |
| *Gateway* and *Router-sensor* | radio ON | $\pm$ 1040$u$ | ≅96<br>*rounds* |

We evaluate SAND along the following metrics: *(i)* the network lifetime and the energy conservation; *(ii)* the forwarding robustness; *(iii)* the sensing fidelity preservation; and *(iv)* the effects of network density.

**Experimental results.** One of the SAND goals it to preserve network routing fidelity. We consider that if paths in the gateway/router-sensor nodes backbone exist, there are similar non-conflicting paths in the underlying network. Fig. 3(a) evaluates the robustness of SAND in ensuring a sink connectivity in a 1,900-nodes network. For this purpose, we vary the sink node density from 0.5%, 1%, and 1.5% of the total number of nodes. As expected, as sink resolution decreases, more active nodes are needed to ensure sink connectivities. This results in a decrease of the network lifetime and the faster energy exhaustion of nodes. Nevertheless, compared to results obtained in Without-SAND topology with 1% of sink density, SAND still extends the number of forwarding paths of network (95% of paths are ensured for a double of time than in Without-SAND), even with a lower sink resolution of 0.5%.

Additionally, for different neighborhood densities, we show that despite using fewer forwarding nodes, SAND does not significantly increase the number of hops of paths to sink nodes. Table 3 shows the average ($\tau$) and the standard deviation ($\sigma$) path length of SAND and Without-SAND, calculated while 90% of the generated stimuli are sensed and corrected forwarded. SAND constructs forwarding paths with only a slightly higher number of hops, on average 18% of more hops.

**Table 3: Path length results.**

| Number of nodes | Neighborhood density | SAND $\tau$ | SAND $\sigma$ | W/o-SAND $\tau$ | W/o-SAND $\sigma$ |
|---|---|---|---|---|---|
| 1,000 | 10 | 3.146 | 1.844 | 2.566 | 1.843 |
| 1,900 | 20 | 3.408 | 2.267 | 2.762 | 2.047 |
| 2,800 | 30 | 3.043 | 2.086 | 2.461 | 2.057 |
| 4,700 | 50 | 2.486 | 2.046 | 1.961 | 2.114 |

Fig. 3(b) shows the average of energy remaining at each node after 40 simulation rounds under different neighborhood densities. In this simulation round, all nodes in the network are still alive. The figure also compares the energy conservation resulted in SAND with the case where no power management is performed. We observe that SAND provides a considerable amount of energy saving over Without-SAND. This is because all nodes in Without-SAND topology have the same energy consumption as router-sensor nodes. In SAND, however, nodes switch when ever possible to sleep or sensor-only states, as a few forwarding nodes are present in each transmission range. We also check that the energy saving increases proportionally to the neighborhood density. This is due to the fact that as the density increases, a lower fraction of alive nodes is composed of forwarding nodes (the highest consumer of energy state), while a higher fraction are in the sleep state (see Fig. 3(e)).

Fig. 3(c) evaluates the robustness of SAND with respect to sensing fidelity, in a 1,900-nodes network. We vary the required sensor node density in the network from 2, 7, and 10 nodes, and show the rate of success sensed stimuli for each density. Compared to the results obtained in Without-SAND topology (where all nodes are sensing for all the simulation time), SAND extends the number of sensed stimuli over the network lifetime, even when a lower sensor node resolution of 2 is used per target area. SAND ensures for a triple of time, 95% of sensing fidelity than Without-SAND. As expected, the increase of the number of required sensor density, impacts the energy consumption of nodes and consequently, decreases the network lifetime.

Fig 3(d) shows the results of the performance of SAND in ensuring sink connectivity for different neighborhood densities (for 20 and 50 neighbors in range) and sink resolution (for 0.5%, 1%, and 1.5% of the total number of nodes). As expected, SAND is less impacted by sink resolution when the neighborhood density increases. This is because the increase of neighborhood density, increases the number of forwarding candidates, and consequently, increases the probability of finding paths to sink nodes.

Fig 3(e) shows the fraction of active and forwarding nodes

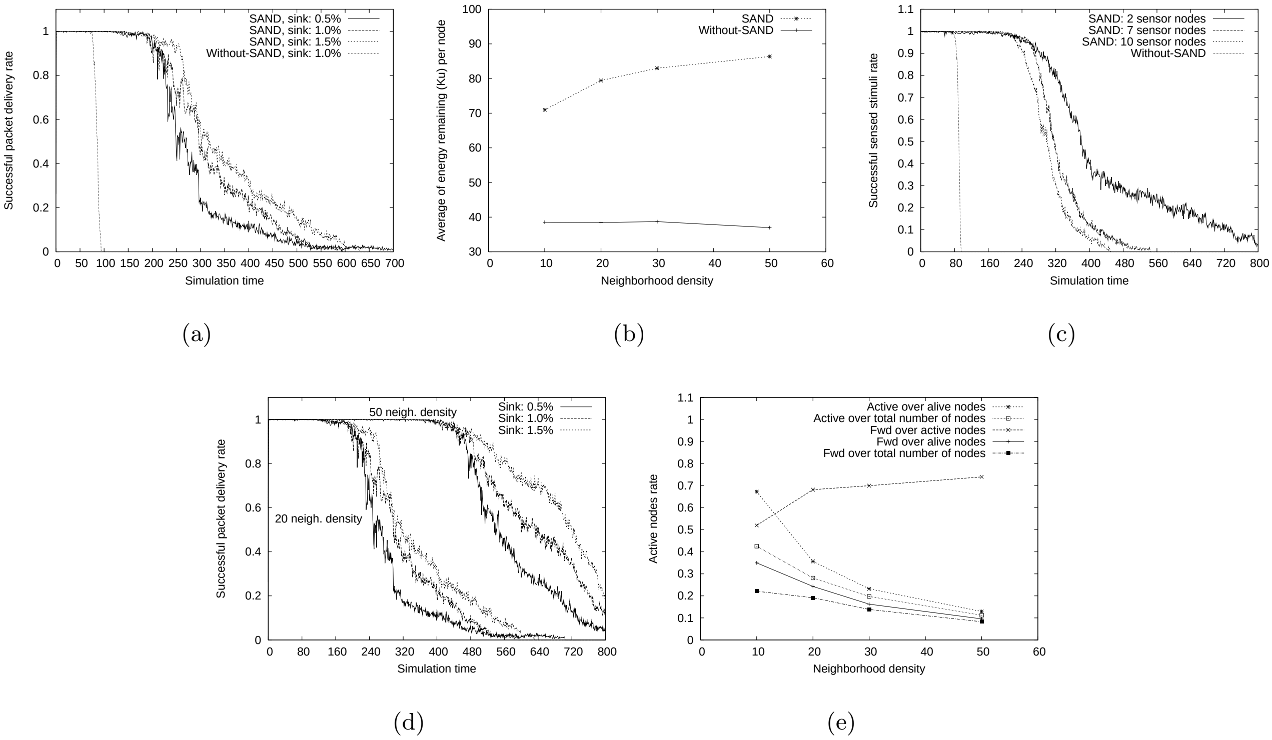


**Figure 3: (a) The sink connectivity. (b) Average of energy remaining after 40 rounds. (c) The stimuli sensing fidelity. (d) Successful forwarding paths. (e) Fraction of active and forwarding nodes.**

after 120 rounds of simulation (among many experiments), under different neighborhood densities. Here, we compare the fraction of active and forwarding nodes over the total number of nodes in the network and over the total number of alive nodes. We show that the potential of saving energy of SAND depends on the node density, since the fraction of active and forwarding nodes depend on the number of nodes per radio coverage area. We can easily verify that the higher the node density, the lower the percentage of active nodes in the network. Also the rate of forwarding nodes over the number of active nodes decreases as the neighborhood density increase. On the other hand, despite the increase of the rate of forwarding nodes over the total number of nodes, the number of forwarding nodes becomes almost constant for higher neighborhood densities (*e.g.*, for 30 and 50 neighbors in range).

# 6. CONCLUSION

In this paper, we have presented SAND, an approach to energy conservation for wireless sensor networks. Energy consumption is one of the most important factors that determines sensor node lifetime. SAND is a fully decentralized, simple, and efficient algorithm able to significantly extend not only sensor lifetime but also the entire network lifetime. SAND focuses on turning off the nodes radio as much as possible while still ensuring stimuli sensing and multi-hop routing fidelity. We have presented the algorithm analysis and the simulation of SAND. Our experiments show that SAND guarantees for a longer time, (1) the existence of paths between any sensor node to at least one sink node in the network and (2) the correct sensing of stimulus in a monitored sensor network. SAND improves considerably network lifetime proportionally to node density, at the price of the slightly increasing paths length from sensor to sink nodes. Additional analyses are being performed to evaluate the network behaviour of SAND as nodes move. We also intend to study network partitions and to validate our results with physical hardware in real scenarios.

# 7. REFERENCES

[1] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–114, Mar. 2002.

[2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 263–270, 1999.

[3] V. Tsiatsis, S. A. Zimbeck, and M. B. Srivastava, "Architecture strategies for energy-efficient packet forwarding in wireless sensor networks," in *Proceedings of ACM International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 92–95, Aug. 2001.

[4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 85–96, July 2001.

[5] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 70–84, July 2001.

[6] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, May 2002.

[7] J. Carle, A. Gallais, and D. Simplot-Ryl, "Preserving area coverage in wireless sensor networks based on surface coverage relay dominating sets," in *Proceedings of IEEE Symposium on Computers and Communications*, pp. 347–352, June 2005.

[8] D. Tian and N. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in

*Proceedings of ACM Wireless Sensor Network and Application Workshop*, Sept. 2002.

[9] J. Carle, A. Gallais, D. Simplot-Ryl, and I. Stojmenovic, "Localized sensor area coverage with low communication overhead," in *Proc. of IEEE Int'l Conference on Pervasive Computing and Communications*, pp. 328–337, 2006.

[10] M. Chatterjee, S. K. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, vol. 5, no. 2, pp. 193–204, Apr. 2002.

[11] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[12] E. L. Merrer, A.-M. Kermarrec, and D. Neveux, "Sonde: Self-organizing density for peer-to-peer network services," tech. rep., IRISA/INRIA-Rennes, France, 2006.

[13] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design & Test of Computers*, vol. 18, no. 2, pp. 62–74, Mar. 2001.