Adaptive Autonomy in Wireless Sensor Networks

Mirgita Frasheri
Mälardalen University
Västerås, Sweden
mirgita.frasheri@mdh.se

José Cano-García
University of Malaga
Malaga, Spain
jcgarcia@uma.es

Eva González-Parada
University of Malaga
Malaga, Spain
gonzalez@uma.es

Baran Çürüklü
Mälardalen University
Västerås, Sweden
baran.curuklu@mdh.se

Mikael Ekström
Mälardalen University
Västerås, Sweden
mikael.ekstrom@mdh.se

Alessandro V. Papadopoulos
Mälardalen University
Västerås, Sweden
alessandro.papadopoulos@mdh.se

Cristina Urdiales
University of Malaga
Malaga, Spain
acurdiales@uma.es

ABSTRACT

Moving nodes in a Mobile Wireless Sensor Network (MWSN) typically have two maintenance objectives: (i) extend the coverage of the network as long as possible to a target area, and (ii) extend the longevity of the network as much as possible. As nodes move and also route traffic in the network, their battery levels deplete differently for each node. Dead nodes lead to loss of connectivity and even to disengaging full parts of the network. Several reactive and rule-based approaches have been proposed to solve this issue by adapting redeployment to depleted nodes. However, in large networks a cooperative approach may increase performance by taking the evolution of node battery and traffic into account. In this paper, we present a hybrid agent-based architecture that addresses the problem of depleting nodes during the maintenance phase of a MWSN. Agents, each assigned to a node, collaborate and adapt their behaviour to their battery levels. The collaborative behavior is modeled through the willingness to interact abstraction, which defines when agents ask and give help to one another. Thus, depleting nodes may ask to be replaced by healthier counterparts and move to areas with less traffic or to a collection point. At the lower level, negotiations trigger a reactive navigation behaviour based on Social Potential Fields (SPF). It is shown that the proposed method improves coverage and extends network longevity in an environment without obstacles as compared to SPF alone.

KEYWORDS

self-organisation; multi-robot systems; networked systems and distributed robotics

ACM Reference Format:
Mirgita Frasheri, José Cano-García, Eva González-Parada, Baran Çürüklü, Mikael Ekström, Alessandro V. Papadopoulos, and Cristina Urdiales. 2020. Adaptive Autonomy in Wireless Sensor Networks. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

Wireless sensor networks (WSNs) consist of a number of interconnected spatially distributed nodes equipped with sensors to capture information from the environment. WSNs are fit for different scenarios, ranging from emergency deployment – where the network needs to be established as fast as possible – to long term monitoring — where the network must last as long as possible, e.g., sensors for irrigation or vibration sensors for seismic areas. In all cases, the network needs to provide an appropriate coverage of the target area. WSNs may be homogeneous, when all nodes have the same role, or heterogeneous. In both cases, WSNs typically include one or more sink nodes, Access Points (APs), to redirect gathered data to external networks. The locations of the APs are typically predefined and stationary, because they are usually connected to a main power supply to route all the network traffic to an external network. However, most nodes in WSNs are typically battery-operated. Battery life depends on connectivity and throughput, which in turn primarily depend the on deployment method [29]. Unfortunately, WSN deployment cannot be optimized in many scenarios and it becomes quite complex for large networks [8]. The locations of the nodes in the network could be computed with optimal approaches. However once the nodes begin to fail, those configurations are no longer optimal. Thus, optimal approaches are not suitable for WSN, since their nodes are prone to failure [25].

Mobile WSN (MWSN) try to solve this problem by providing nodes some degree of mobility, so that they can deploy themselves (self-deployment) or, at least, adjust their locations (self-healing) when other nodes start to fail, in order to cover the gaps that appear in the network [36]. Although there are other options, Multiple Robot Systems (MRS) have been often used in combination with MWSN to transport nodes when needed [20, 21, 25].

As in traditional autonomous robot navigation problems, there are deliberative and reactive approaches for MWSN navigation. Deliberative approaches are meant to optimize efficiency [11, 18, 23–25, 35], however, they require a reliable model of the environment – including network configuration, environment layout, traffic, etc. – and they are computationally expensive. Reactive approaches rely on local factors instead [5, 7, 22, 27, 37], e.g., setting a local behavior

for each node or a set of local rules that the nodes must follow. The combination of all local behaviors provide an emergent global behaviour with reduced computational cost, where results are not optimized and local minima may appear. In order to avoid the drawbacks mentioned above, in the robotics domain it is common to combine both deliberative and reactive strategies into a hybrid approach.

A reactive approach to MWSN deployment and self-healing was proposed in [17, 32]. The proposed approaches are a variation of the Social Potential Fields (SPF) [28], originally proposed for navigation in a robot swarm. The underlying idea is that each robot is affected by a number of forces, namely: (i) a repulsion force depending on the Received Signal Strength (RSS) that leads to network dispersion; (ii) an attraction force depending on how many nodes each one is linked to that preserves connectivity; and (iii) a repulsion force depending on nearby obstacles that avoids collisions. Nodes in the MWSN move until the sum of these forces is equal to zero, i.e., the network is balanced. Then, they stay at their locations until a number of nodes run out of battery and the network needs to be balanced again. This algorithm successfully extended the network life preserving coverage as much as possible and outperformed rule-based systems like the Backbone algorithm [9]. However, given its reactive nature, it had a number of problems because no high level strategy was used to determine specific node locations. In this work, we propose to use an agent architecture to add a cooperative layer to the reactive system.

Agents have already been used in MRS, specifically in robot teams and swarms [4, 31, 34]. Coalition formation algorithms designed for multi-agent systems have been adapted for the multi-robot domain in order to solve tasks such as box-pushing, cleaning, and sentry duty [33, 34]. Cognitive agents have been integrated into robotic swarm architectures with the purpose of guiding the behaviour of the group without losing the advantages of the swarm approach [4]. Others have applied distributed scheduling mechanisms in robot teams operating in a hospital environment, mapping an agent to a robot, thus allowing each robot to compute its own schedule in a distributed way [31].

Another relevant aspect relates to the autonomy of agents involved in the decision-making. Autonomy can be defined based on the dependencies between agents, i.e., when an agent a needs the intervention of another agent b in order to complete its goal G, then a depends on b for G, and is not autonomous with respect to b in this context [6]. Furthermore, mechanisms with which agents change their levels of autonomy based on the circumstances, have also been investigated. They range from adjustable autonomy, where there is a human in the loop, an operator, which makes the decision on whether to change autonomy levels of the agents, to adaptive autonomy, where agents themselves decide their own level of autonomy [19].

In this work we present a hybrid control architecture for a swarm of robot MWSN nodes where the reactive layer is controlled by a SPF and the cooperative layer relies on an agent architecture that allows agents to adapt their autonomy during their operation. This agent approach is based on the willingness to interact abstraction, which determines when agents ask and give help to one another [12]. As in most works involving a large number of robotic entities [21, 25], the evaluation of our hypothesis – that the hybrid agent approach yields better performance than SPF alone – relies on simulations. However, the working parameters for the presented algorithm have been set using a reduced number of real robots. Results show that the proposed method increases the coverage in the network, as well as the network longevity as compared to SPF alone. Furthermore, the achieved improvement is consistent for the whole operation of the network.

The rest of this paper is organized as follows. Section 2 describes in detail the problem to be solved. Section 3 presents the previously proposed low level reactive navigation algorithm. Section 4 describes the hybrid agent approach and the willingness to interact abstraction that affects how agents collaborate with one another. Section 5 describes the design of the experiment, while Section 6 presents our results. Concluding remarks and an outline of directions for future work are given in Section 7.

2 PROBLEM FORMULATION

Let's assume an area Z with known dimensions $\langle d_w, d_h \rangle$, and n agents[1] $a_i \in \{a_1, \ldots, a_n\}$ allowed to move in Z. Each agent is characterized by its battery level b_i, and it is able to connect to others within its communication range r_C. An agent's battery is impacted by: (i) its motion in Z; and (ii) the amount of traffic it routes. Therefore, agents will have different life-spans, depending on how much distance they have covered, and how much traffic they have routed. The amount of traffic depends on the topology of the network and on the routing algorithm, thus, it is very hard to predict a priori in MWSN. Similarly, the robot motion in a configuration with a large number of agents is also hard to predict. Instead, these parameters are measured by each node through its life.

Agents have two goals, (i) to provide as much coverage for the network as possible, and (ii) to extend the longevity of the network as much as possible. Coverage depends on the network topology, so it is important to distribute robots adequately. The longevity of the network depends on the lifespans of the robots in the MWSN. Therefore, the second goal is addressed by both minimizing the travelled distance of the agents, which aims at potentially extending their lifespans, and by distributing the traffic load uniformly among nodes.

3 SPF

As commented, the reactive system consists of a cooperative agent based layer and a reactive low level algorithm. Specifically, the reactive layer uses SPF algorithm proposed previously [32] for deployment and self-healing. The original SPF was proposed for autonomous navigation in swarm robots by Reif and Wang [28]. SPF defines attraction and repulsion forces to set goals and/or constraints. In our system, our goals are: i) to spread the network to achieve the maximum possible coverage with living nodes; ii) to preserve connectivity among living nodes; and iii) to avoid collisions when the robots move. Hence, three forces are defined.

- Obstacle repulsion force(s) $f_{r1}(r_{i,j})$ repel the robot from other robots or physical obstacles in its vicinity to prevent collisions. This force is only significant in the vicinity of physical objects.

[1] In this paper the terms agent, robot, and WSN node are used interchangeably, as we have one agent per robot.

- Deployment repulsion force(s) $f_{r2}(r_{i,j})$ moves robots away from each other to spread them and increase coverage.
- Cohesion force (connectivity attraction) $f_c(r_{i,j})$ increases with $r_{i,j}$ to avoid loss of communication among nodes.

$r_{i,j}$ being the distance between robots i and j. In order to estimate relative distances between robots and with respect to the coverage area boundaries, we need a node localization technique. The locations of all elements under simulation is always known. In real tests, locations could be estimated by combining robot odometry and RSS-based trilateration [1, 26].

The combination of all three forces for each node in the network returns an emergent motion vector. A node stops moving when that vector is under a threshold f_u. When all nodes stop, the network is balanced. Afterwards, if any force changes for a node, e.g., nearby nodes die or they have to move significantly for one reason or another, that node moves again. If it affects other nodes in the region, the network will need to be balanced again. This process is repeated in time until the performance of the network is deemed insufficient after enough nodes have died.

It can be observed that there is no high level strategy in the robot motion at this point. Nodes simply respond to the defined forces and no plan is made to increase the network life by considering the specifics of each node. The main novelty of the present work is the addition of an agent layer over the SPF. We assign an agent to each robot in the network, which is aware of the different parameters that the robot uses to calculate SPF forces, in addition to its battery level and routed traffic. Agents may decide that some robots need to switch location with others when their battery level is not fit to route the traffic in the area. Thus, by purposefully redirecting nodes when necessary to adapt to the network traffic specifics, we increase its average life. The following sections describe this proposal in detail.

4 AGENT APPROACH

This section describes how a cooperative agent approach based on the willingness to interact is combined with the presented reactive SPF method, in order to address the problem described in Section 2 for maximising the coverage and extending the longevity of a wireless sensor network. A detailed account is given on how agents compute their willingness to interact, and on the negotiation protocol used.

4.1 Agent Behaviour in the MWSN

The operation of agents in a wireless sensor network follows three phases, initialization, deployment, and maintenance. During the first phase agents are started and their state variables are initialized. Afterwards, the deployment phase follows using SPF, in which agents move away from the initial locations in order to extend coverage of the target area as much as possible, while keeping the connectivity to the AP. When the deployment is complete and the network is stabilized, agents continue draining their batteries due to traffic routing. As commented, not all agents have the same battery level after deployment, as they have not travelled the same distance. Furthermore, depending on their location and the routing algorithm, some may route far more traffic than others. Hence, some agents may be depleted before others. If an agent is depleted, it may stay on location as a new obstacle or travel back to a battery charging station, leaving a "hole" in the network. When a number of agents have died, the SPF needs to be run again to re-stabilize the network. Alternatively, rather than waiting for nodes to be depleted, the proposed approach gets nodes with higher battery levels to replace the ones that are about to be depleted. Thus, allowing for a more graceful degradation of the coverage, and extension of the life of the network.

Let's assume a critical battery level b_{l0} defined as the amount of battery that an agent needs to go to a collection point, e.g., an AP. This value could be different for every agent, as it depends on where in the network an agent is positioned, or we could have a safe threshold that guarantees that a node can reach the collection point from any location in the coverage area for simplicity. Furthermore, let's assume two battery levels b_{lh} and b_{l1} defined as $b_{l0} \leq b_{lh} \leq 10\% \cdot b_{l0} + b_{l0}$, and $b_{l0} \leq b_{l1} \leq 30\% \cdot b_{l0} + b_{l0}$, respectively. The thresholds are selected heuristically and are used to indicate how close a robot's battery is to its critical level.

The first agent to reach the b_{l0} triggers the first negotiation round in the network. Then, all agents with battery level below $b_{l1}^{(t)}$ start sending help requests to the network to ask agents with battery above $b_{l1}^{(t)}$ to replace them. The urgency of a request sent by an agent in need, as well as the disposition to help by other agents is captured in the willingness to interact, as described in the next subsection. The agents with battery above $b_{l1}^{(t)}$ are the ones that will respond to these help requests and move to new locations if so decided.

After a negotiation round is complete, all agents with battery level below $b_{lh}^{(t)}$ move towards their collection point. These agents are too depleted to be useful anymore. Note that we do not remove only the one node that reached $b_{l0}^{(t)}$, but a percentile of nodes that will reach it in the near future. Since most nodes will probably move during the balancing stage, nodes with the lowest battery would probably reach the removal threshold shortly after. Hence, setting two thresholds avoids running SPF each time a single node needs to leave. After the assignment to locations is complete, negotiation-winning agents move towards their target positions. Nodes with battery between $[b_{l0}^{(t)}, b_{l1}^{(t)}]$ do not move at all. During this motion stage, only obstacle repulsion forces are active to avoid collisions. Finally, after all moving agents have reached their target locations, the SPF is run again in order to balance the network.

4.2 Willingness to Interact

The willingness to interact defines a general disposition of an agent to interact with other agents, by either asking or giving help [12]. The willingness at time t, $w^{(t)} \in [-1, 1]$, depends on the internal state of an agent, and is not related to any particular task or help request that may have been received from others. In this paper, the calculation of the willingness to interact is influenced only by the current battery level b_c. Note, further, that the previously proposed equation [12] has been adapted in order to include WSN related

parameters. The willingness to interact at a time t is given by,

$$w^{(t)} = \begin{cases} -1, & \text{if } b_c^{(t)} \leq b_{l0}^{(t)}, \\ -\frac{b_c^{(t)} - b_{l0}^{(t)}}{b_c^{(t)}}, & \text{if } b_{l0}^{(t)} < b_c^{(t)} \leq b_{l1}^{(t)}, \\ \frac{b_c^{(t)} - b_{l1}^{(t)}}{b_c^{(t)}}, & \text{if } b_{l1}^{(t)} < b_c^{(t)}. \end{cases} \quad (1)$$

This equation states that when an agent's battery is below $b(t)_{l1}$, the corresponding willingness will be negative, and the agent is in a state where it is ready to ask for help. Furthermore, the closer $b_c^{(t)}$ is to $b_{l0}^{(t)}$, the more negative the willingness will be. If the value for the willingness goes below the threshold $w_H = -\frac{b_{lh}^{(t)} - b_{l0}^{(t)}}{b_c^{(t)}}$ corresponding to $b_{lh}^{(t)}$, the agent might soon reach the critical battery level. When at least one agent's willingness value becomes -1, i.e., at least one has reached the critical battery level, then agents below w_H start asking for help and moving toward the collection point, whereas agents with $w_H \leq w \leq 0$ stay where they are and start asking for help. If $b_{l1}^{(t)} \leq b_c^{(t)}$ then an agent's willingness is positive, therefore the agent can respond to help requests from others in the WSN.

Once a request for help is initiated, agents with positive willingness will reason to meet this request. Furthermore, all requests for agents below w_H are considered, whereas for the others only those requests where the willingness is not more than 20% over w_H are considered. This threshold is also set heuristically, and is used to indicate which agents are closest to the hysteresis level. The willingness specifies the overall disposition of an agent to interact based on its own state. Nevertheless, when an agent gets a request for help, the willingness needs to be refined to reflect the trade-off between staying in the current position with a particular traffic load, and moving to a new position with another traffic load. This adjustment is done by incorporating in the willingness the utility of an agent for moving to a new position with a known traffic load (Equation 2).

$$W^{(t)} = w^{(t)} + u^{(t)}, \quad (2)$$

where $u^{(t)}$ is the utility for moving to a new position, calculated as,

$$u^{(t)} = \begin{cases} 1 - \frac{b_m^{(t)} + b_n^{(t)}}{b_c^{(t)}} + \frac{\frac{b_c^{(t)}}{b_A^{(t)}} - \frac{b_c^{(t)}}{b_B^{(t)}}}{\frac{b_c^{(t)}}{b_A^{(t)}} + \frac{b_c^{(t)}}{b_B^{(t)}}}, & \text{if } 1 - \frac{b_m^{(t)} + b_n^{(t)}}{b_c^{(t)}} \geq 0.3, \\ -w^{(t)}, & \text{otherwise}, \end{cases} \quad (3)$$

where $b_m^{(t)}$ is the battery required to move to the new position, $b_n^{(t)}$ is the battery level to go from the new position to the collection point, $b_A^{(t)}$ is the battery spent on routing per iteration in the current position, and $b_B^{(t)}$ is the battery that would be spent for routing in the new position. This means that, if an agent has at least 30% of battery on top of what is needed to go to a new position, and the collection point thereafter, then it proceeds with reasoning on whether to help. Otherwise, the utility is set to $-w^{(t)}$ and, given Equation 2, the willingness of the agent will be 0, thus it will not take part in the negotiation. When an agent has enough battery for useful motion, then the impact of the traffic load is also considered. If the new position requires more battery, then the utility is decreased, otherwise it is increased. In brief, agents with more battery will favor more traffic intense locations and vice-versa.

In this paper, the willingness to interact is only dependent on the battery level, however, in other applications there might be other state variables to consider, e.g., number of tasks needed to be achieved concurrently. The same consideration holds for the calculation of the utility.

4.3 Negotiation Protocol

The negotiation protocol is triggered when at least one agent in the system reaches its critical battery level b_{l0}, or when the *AP* identifies a dead node in the network that did not ask for help in time. In the former case, agents with negative willingness will send a help request to all alive nodes in the network, collect the responses, assign the agent with the highest positive willingness, and notify the assigned agent by sending a packet in the network. In the latter case, the *AP* will send a help request to all alive agents on behalf of the dead node. Moreover, the AP handles only one dead dead per negotiation round. Afterwards, from the responses, the agent with the highest positive willingness is assigned to the location of the dead robot, and notified. Agents with positive willingness will process the requests and send their answers back to the requesting agents. A request will be considered if it comes from those agents below w_H, as well as those agents with willingness not more than 20% on top of w_H.

All additional packets generated due to the negotiation between agents, and how they affect the battery drainage, are computed. For the calculation of the number of packets that flow through the network and are routed by the nodes, we assume that the network uses a limited flooding strategy known as geographic routing, according to which each packet sent by an origin towards a destination is re-transmitted by other nodes but only if they are nearer to the destination. This type of strategy is widely used in WSN when the network deployment is not planned and signaling traffic needs to be very limited to extend the network life as much as possible [14, 25]. More details are provided in Section 5.2.

5 EXPERIMENT DESIGN

This section describes the hypothesis investigated in this paper, the metrics used for the evaluation of our approach and its comparison with SPF, the communication model, and the simulation setup used for generating the data used in the analysis.

5.1 Hypothesis and Evaluation Metrics

The hypothesis investigated in this paper is formulated as follows:

Hypothesis 1. *The hybrid approach that combines agent collaboration with a reactive layer for adaptation, yields better results with respect to network coverage and longevity of the network in the maintenance phase of a MWSN, as compared to a solely reactive approach.*

In order to evaluate the hypothesis, the following metrics are defined: blanket coverage, which addresses the coverage concern, and energy efficiency and consumption which indirectly address the longevity of the network. Blanket coverage refers to any point of the area of interest covered by at least one node [13]. Assume a node i that covers an area A_i. Then the coverage for n nodes over

the whole area A is calculated as:

$$C = \frac{\bigcup_{i=1,\ldots,n} A_i}{A}. \quad (4)$$

Equation 4 is transformed into a probabilistic model [16] as follows:

$$C = \sum_{i=1}^{m} \frac{p_i}{m}, \quad (5)$$

where p_i is the probability of detecting an event in cell i, in a probabilistic grid with m cells. The probability is given by:

$$p_i = 1 - \overline{p}_j = 1 - \prod_{j=1}^{n}(1 - p_{ij}), \quad (6)$$

where p_{ij} is the probability that node j detects an event at location i.

Aspects that relate to energy are captured on one hand by the uniformity of the network U for n nodes (Equation 7).

$$U = \frac{1}{n}\sum_{s=1}^{n} U_s, \quad (7)$$

$$U_s = \sqrt{\frac{1}{K_s}\sum_{j=1}^{K_s}(r_{s,j} - \overline{r}_s)^2}, \quad (8)$$

where $K_{s,j}$ represents the number of nodes close to node s, $r_{s,j}$ the distance between nodes s and j, and $\overline{r}_s$ the average distance between s and its closest neighbours.

On the other hand, such aspects are also reflected in the average power consumed by the nodes to send a message $\overline{P}$ (Equation 9).

$$\overline{P} = \frac{1}{n}\sum_{j=1}^{n} \overline{P}_j, \quad (9)$$

$$\overline{P}_j = \frac{1}{n}\sum_{j=1}^{n-1} P_{s,j}, \quad (10)$$

$$P_{s,j} = P_{s1} + \cdots + P_{sk}, \quad (11)$$

where $\overline{P}_j$ is the average power for node j to send a packet to the network, and $P_{s,j}$ is the power consumed to send a message from node s to j, over k hops needed for the packet to reach the destination.

Finally, the overhead produced by the packets generated due to the negotiation between agents is estimated and compared to the total number of packets routed in the network.

5.2 Simulation Setup

The hypothesis has been evaluated in simulation[2], by comparing the performance of SPF alone [17], with the proposed hybrid approach combining both SPF and agent collaboration. The environment used for the evaluation of the SPF approach, proposed previously [32], has been extended to support agent negotiation, and relies on the tools provided by the Player/Stage simulator [15].

The simulated area has a size of $150x150\text{m}^2$, scale $1:10$, and is populated by $n = 100$ robots. Each robot is initiated with a battery level $b_0 = 3000\text{mAh}$. Only the sink is assumed to be connected to the power supply, and will not deplete. We have considered two main causes of battery drainage: the routing of packets through the network and the movement of the robots.

Assuming that the robots move at a constant speed, the charge drained from the battery (in $\text{mA}\cdot\text{h}$) when a robot moves 1m can estimated as:

$$b_{1m} = \frac{m_C}{v \cdot 3600} = 1.11\text{mA}\cdot\text{h/m}, \quad (12)$$

where $m_C = 200\text{mA}$ is the electric current that flows through the motor while the robot is moving, and $v = 0.05\text{m/s}$ is the robot speed, in accordance with the consumption model of the Hexbug robot toys derived in previous work [17].

The other cause of battery drainage is the network operation, which is heavily influenced by the nature of the data traffic and also by the behaviour of the communication protocols, and in particular by the routing strategy and the medium access control (MAC) mechanism. A typical low-power and short-range communication transceiver requires a supply current of 10-20mA during its transmission and reception operations [30]. Therefore, each time a node sends a packet some battery charge is consumed, but also the battery is drained when reception is enabled (either if data is actually being received or not). The MAC strategy of the link layer determines how much time the transceiver spends in each possible state (transmission, reception or sleep) during its operation. In order to save battery, a duty-cycling MAC strategy is typically used to allow the transceiver to remain in a low-power state most of the time [3] while maintaining the network operational. Thus, our simulation model considers two main sources of battery consumption due to network operations: a constant background consumption due to the duty-cycling operation, which is equal for every node in the network, and the consumption caused by packet transmissions, which depends on how many packets each node is sending and/or routing.

Assuming that the MAC parameters are properly tuned, the average current drained during duty-cycling operation can be as low as 1% of the current required when the transceiver is in the receiving state [10], so in our simulation model we have considered that $0.15\text{mA}\cdot\text{h}$ are drained from the battery each hour of network operation. On the other hand, duty-cycling strategies require each packet to be repeatedly transmitted over a period of time to guarantee it is finally received, which increases the battery waste of sending a packet. In our simulation we assume a wake-up frequency of 20 Hz for the duty-cycling protocol, so each packet should be transmitted repeatedly during a 50ms interval. The supply current of the transceiver during the transmission operation depends slightly on transmission power, which in turn determines transmission range. In accordance with the datasheet of a commercial transceiver [30], our model considers a 12.5mA supply current during transmission for a coverage range of 25m. Regarding all this, the charge drained from the battery each time a packet is sent and/or routed by a node is set to $1.74 \cdot 10^{-4}\text{mA}\cdot\text{h}$. To calculate the amount of packets routed by each node, both the data traffic sent by the sensor nodes to the AP and the packets sent during the agent negotiation rounds have to be regarded. A simple model has been considered for the data traffic sent by the nodes to the AP during network operation: each node periodically tries to send one packet to the AP (at a rate of one packet every 10 seconds), which is routed through the network by intermediate nodes. Also, during the agent negotiations,

[2]The code for running the simulations is publicly available at https://bitbucket.org/gitagent/gitagent_wsn/src/master/

some packets have to be exchanged between nodes requesting help, nodes willing to help, and the AP, and these packets are also routed by intermediate nodes in order to reach their destinations. As mentioned in Section 4.3, a geographic routing mechanism is adopted to compute the number of packets each nodes routes. Geographic routing operates over multi-hop mesh topologies and its main advantages when compared to other routing strategies for WSN are that it provides a reasonable scalability, adapts quickly to network topological changes, and is stateless and, therefore, does not require additional signaling for routing [2, 14]. Our simulation model implements one of the simplest approaches to geographic routing, the closest neighbor routing: a given node i broadcasts a packet that is received by all of its neighbor nodes within its transmission range. The receiving neighbor nodes broadcast the packet to their own coverage zone, but only if they are closer to the destination node than the one from which they received the message. A node never re-transmits the same packet more than once. This algorithm generates a partial flooding, but is very simple and reasonably energy-efficient.

In regards to robot motion, the parameters of the three different forces applied to each robot except the AP, have been obtained heuristically, and depend on the modeled physical robots. The obstacle repulsion force is calculated as:

$$f_{r1}(r_{i,j}) = -\frac{0.001}{(r_{i,j})^8}. \tag{13}$$

The deployment repulsion force is calculated as:

$$f_{r2}(r_{i,j}) = -\frac{20}{(r_{i,j})^7}. \tag{14}$$

The cohesion force is calculated as:

$$f_c(r_{i,j}) = -\frac{\alpha \cdot n_{fail}}{n \cdot n_{nearAP}}, \tag{15}$$

where n_{fail} is the number of depleted nodes, and n_{nearAP} is the number of nodes directly connected to the AP (i.e., located within the coverage area of the AP).

The simulation will stop when more than 70% of the robots have depleted their battery, i.e., their current battery level is lower than 5% of the initial battery level. It is assumed that the locations of all the robots, as well as the boundaries of the area are known. In a physical environment the robots would rely on triangulation using their own RF signals for localization. Ideally, the sink node will be static as well, so its position will be also known. Usually, at least two more beacons (with known positions) are used in the deployment area so the other nodes can triangulate themselves. Of course, triangulation errors may be up to a few meters and the error will also propagate for nodes that triangulate themselves using information from mobile nodes instead of beacons. Nevertheless, the error will not accumulate over time, as the positions are recalculated from the RSS at every new triangulation. More importantly, these errors are not critical for the application – at most, robots will end a few meters away from the destination or re-transmit some unnecessary packets. If localization is critical, more beacons could be added where necessary. Finally, there are no obstacles in the environment, but dead robots with depleted batteries are treated as obstacles.

Table 1: Statistics for the metrics over 30 runs.

	Statistics for 50% nodes dead							
	cov.		dist.(m)		uniform.		time(days)	
	avg	std	avg	std	avg	std	avg	std
$SPF1$	0.45	0.008	84.44	1.32	0.53	0.0056	65.84	1.85
$SPF2$	0.02	0.001	49.81	0.56	0.43	0.0110	106.83	5.96
$AA1$	0.48	0.012	94.44	4.55	0.51	0.0074	63.89	2.04
$AA2$	0.67	0.021	80.13	3.44	0.71	0.0152	75.77	3.37
	Statistics at simulation end							
	cov.		dist.(m)		uniform.		time(days)	
	avg	std	avg	std	avg	std	avg	std
$SPF1$	0.30	0.0088	102.23	2.05	0.45	0.0091	110.2	4.6
$SPF2$	0.30	0.2435	62.76	2.12	0.54	0.1118	209.0	10.9
$AA1$	0.28	0.0050	144.95	8.62	0.39	0.0113	108.7	2.7
$AA2$	0.41	0.0097	109.73	7.53	0.55	0.0298	139.5	5.5

6 RESULTS

The comparison between the combined agent approach and the SPF was done by considering two values for the α parameter in the calculation of the cohesion force $f_c(r_{i,j})$ active in the network, $\alpha \in \{5, 20\}$. Thus, there are two configurations for each approach, namely (i) $SPF1$ with the base-line value for the cohesion force defined as in previous work ($\alpha = 20$) [17, 32], (ii) $SPF2$ with reduced cohesion force ($\alpha = 5$), (iii) $AA1$ with the base-line cohesion force, and (iv) $AA2$ with reduced cohesion force. Note that, for $AA1$ and $AA2$, the battery waste for the packets generated due to the negotiation is taken into account in the battery consumption of each robot.

Simulations for each of the four cases were repeated 30 times, with the corresponding means and standard deviations (see Table 1) taken over the values of each metric at the step where less than 50% of the nodes are alive, and at the end of the simulation. Results in Figures 1-7 consist of the curves for each simulation run, as well as the average over 30 runs, for every method. Visually, methods can be distinguished by the color hue, with bolder lines for the averages. A direct average of the curves is not possible because each run yields different length of the simulation. The average is therefore computed considering up to the shortest simulation for the considered method.

The averaged values (Table 1) show the state of the network under the different methods for two different time points as aforementioned. It can be seen that when the 50% mark for the dead nodes is reached, $AA2$ achieves better coverage and uniformity with respect to the rest, whereas with respect to the distance walked it performs better than $AA1$. Moreover, $SPF2$ is the last to reach the 50% mark, followed by $AA2$, with $SPF1$ and $AA1$ being the fastest to reach such level of dead nodes. At the end of the simulation, $AA2$ maintains higher levels of the coverage, with the rest of the methods showing slightly worse performance. Regarding uniformity, at the end of the simulation $AA2$ and $SPF2$ are rather comparable, outperforming the other two. In terms of walked distance and longevity of the network, the results are similar to the results for the 50% mark. During the operation of the network, as time passes and nodes become depleted, the $AA2$ method consistently achieves

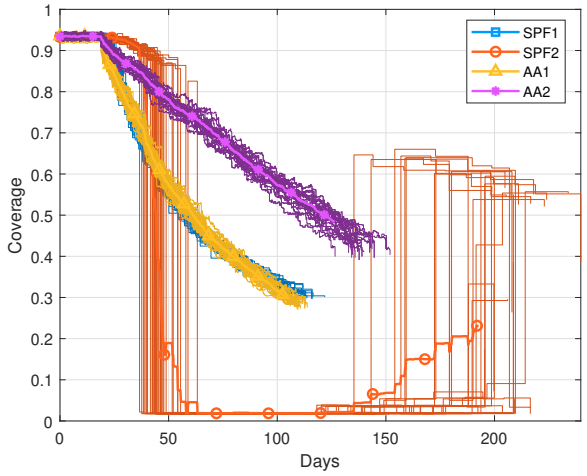


Figure 1: Coverage with respect to time

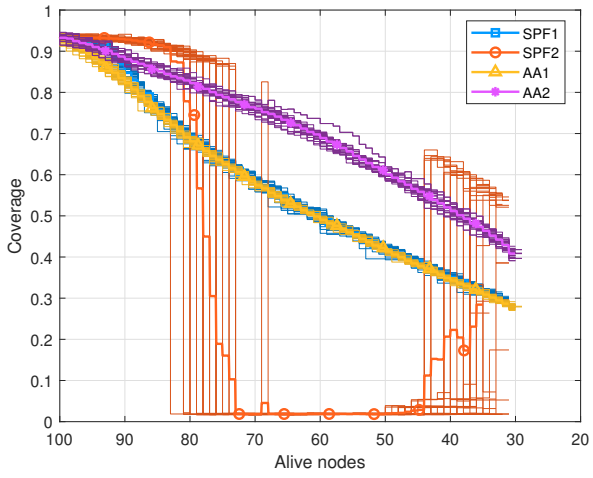
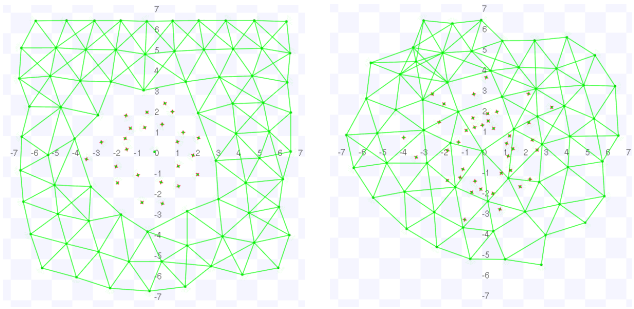


Figure 2: Coverage with respect to the number of alive nodes



a) SPF2 b) AA2

Figure 3: Robot layout on the simulated map for an intermediate step of the simulation.

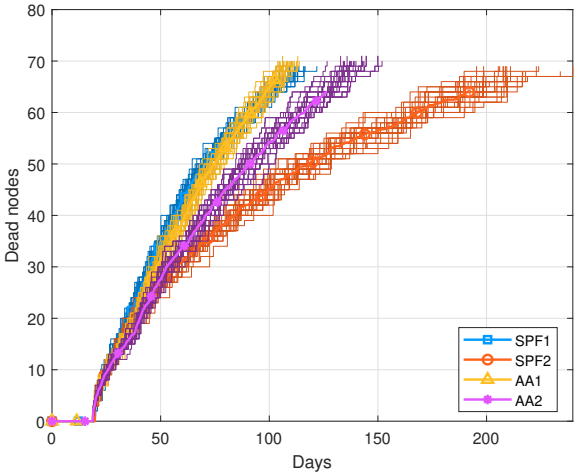


Figure 4: Depleted nodes with respect to time

higher coverage than the others (Figures 1-2) [3]. Note that an area A_i is considered covered if the covering node is connected to the AP, either directly, or through several hops. The nodes close to the AP route higher amounts of traffic, as compared to the nodes in the outskirts of the network. Thus, they become depleted faster. When the cohesion force is reduced for *SPF2*, and the central nodes start to die, the network does not shrink to keep the connectivity to the AP. Instead a gap is created in the network (see Figure 3). The resulting coverage goes close to 0. Nevertheless, it can happen that the nodes manage to come closer to the AP and reconnect at the end of the simulation, thus improving on the coverage as well as the other metrics. In the case of *AA2*, when central nodes start to die, they are replaced by nodes at the outskirts, as such the nodes remain connected to the AP. The nodes at the outskirts are the first to move because their battery levels are highest, which is due to the low traffic load. It is possible to observe that the *AA2* method yields consistently better coverage, throughout the whole operation time of the network, oppositely to *SPF2* which can by chance recuperate close to the end of the simulation.

The network, in terms of alive nodes (Figure 4), lasts the longest for the *SPF2* method. However, in this case alive nodes become disconnected from the AP. The *SPF1* and *AA1* methods are comparable in terms of depleted nodes over time, nevertheless with *SPF1* nodes last longer. The *AA2* method has a consistently lower rate of depleted nodes over time as compared to *SPF1* and *AA1*, resulting in highest network longevity. The life of the network is extended with circa 29 days on average as compared to *SPF1*. With respect to the distance traveled by nodes (Figure 5), *SPF1* and *AA2* are comparable, whereas *AA1* has the highest distance traveled. The *SPF2* method has the lowest traveled distance, due to the reduced cohesion force, the nodes will not move towards the AP to keep the connectivity.

The *AA1* and *AA2* methods generate additional packets in the network due to the negotiation between agents overtime. The averages for the total amount of negotiation packets over the total amount

[3]Note that the sampling frequencies for the four methods is different. This is because the SPF-based methods are only rerun when the nodes breakdown and the network should balance. In between, the alive nodes are stationary. However, for the agent-based methods, the negotiations start taking place before any nodes breakdown, i.e., the time between failures of nodes is simulated as well.

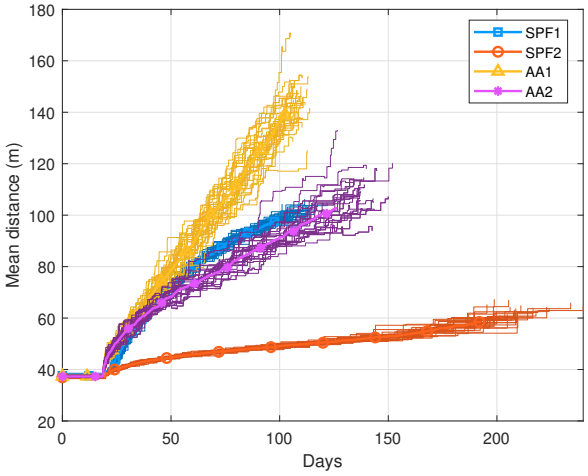


Figure 5: Distance traversed with respect to time. Note that the y-axis starts from 20 meters

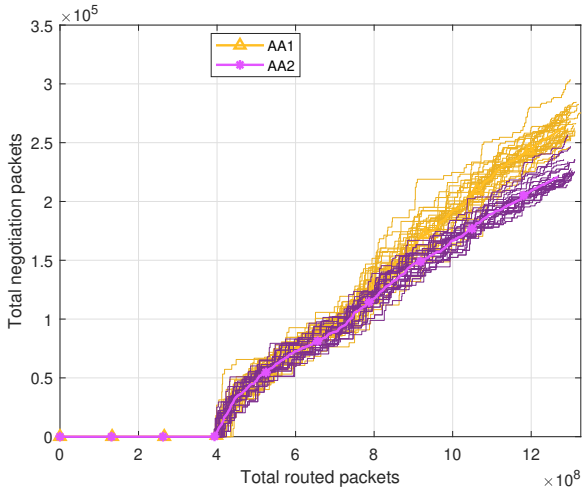


Figure 6: Number of negotiation packets vs the total number of routed packets.

of routed packets in the network are 2.1×10^{-4} and 1.8×10^{-4}, for *AA*1 and *AA*2 respectively (Figure 6). Therefore, the battery consumption due to this overhead is negligible.

The *AA*2 method outperforms the other methods in our tests with respect to uniformity (Figure 7). Whereas, *SPF*1 and *AA*1 are comparable until the 65^{th} day, approximately. Afterwards, as the nodes keep dying, uniformity is preserved better with *SPF*1. Similarly to the coverage metric, for some experiments *SPF*2 regains connectivity close to the end of the simulation, thus improving on the uniformity as well.

7 CONCLUSION

A hybrid agent approach, which combines a reactive layer with explicit collaboration between agents for the self-healing phase in mobile wireless sensor network has been presented. The reactive

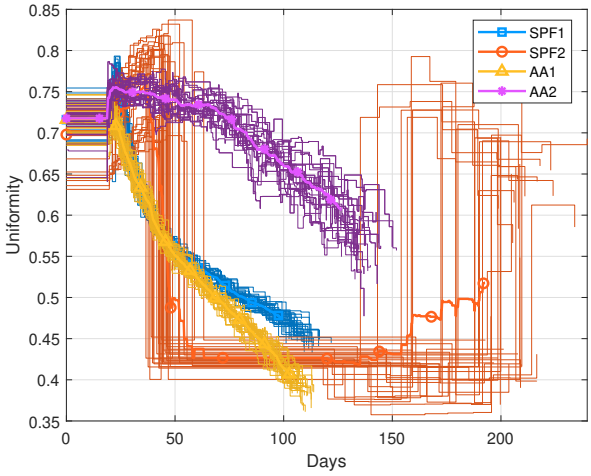


Figure 7: Uniformity in the network with respect to time. Note that the y-axis starts from 0.35 units.

layer is based on a SPF algorithm. The agent collaboration part is modeled through the willingness to interact abstraction, which defines when agents ask and give help to each other based on their battery level. It is shown that the hybrid agent approach improves the coverage and longevity of the network, as compared to the social potential fields algorithm. Furthermore, the proposed approach does not require a high cohesion force to keep the nodes connected to the AP. Instead, this comes as a result of how agents negotiate with one another. After a negotiation, nodes at the outskirts of the network with low traffic load, take the place of central nodes, which become depleted at a faster pace.

There are four lines of inquiry for future work. Firstly, the comparison between the hybrid agent approach and the social potential fields algorithm can be extended to account for environments with obstacles. Secondly, machine learning techniques can be used to adjust the forces for each node in the network, depending on individual battery levels, and traffic load. Thirdly, other delegation strategies could be investigated, and compared to the current work where agents drop their current positions when assigned to new ones. Furthermore, the reassignment of agents could be negotiated before the critical battery level is reached, in order to send depleting agents in locations with less traffic. Lastly, it is of interest to evaluate the proposed approach for hierarchical networks with several levels of the nodes.

ACKNOWLEDGMENTS

The work in this paper was funded by the VINNOVA project UNICORN, the DPAC research profile (KKS funding 20150022), by the FIESTA KKS project, by the Swedish Foundation for Strategic Research under the project “Future factories in the cloud (FiC)” with grant number GMT14-0032, and by the Spanish project RTI2018-096701-B-C21 and the PY18-1652 Andalusian regional project.

REFERENCES

[1] K Aamodt. 2006. CC2431 location engine. *Application Note AN042 (Rev. 1.0), SWRA095, Texas Instruments* 2 (2006), 2–4.

[2] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. 2005. Wireless Mesh Networks: A Survey. *Comput. Netw. ISDN Syst.* 47, 4 (mar 2005), 445–487. https://doi.org/10.1016/j.comnet.2004.12.001

[3] Fayez Alfayez, Mohammad Hammoudeh, and Abdelrahman Abuarqoub. 2015. A Survey on MAC Protocols for Duty-cycled Wireless Sensor Networks. *Procedia Computer Science* 73 (2015), 482 – 489.

[4] F Aznar, Mireia Sempere, FJ Mora, Pilar Arques, JA Puchol, Mar Pujol, and Ramón Rizo. 2011. Agents for swarm robotics: Architecture and implementation. In *Highlights in Practical Applications of Agents and Multiagent Systems*. Springer, 117–124.

[5] Novella Bartolini, Tiziana Calamoneri, Emanuele Guido Fusco, Annalisa Massini, and Simone Silvestri. 2010. Push & Pull: Autonomous deployment of mobile sensors for a complete coverage. *Wireless Networks* 16, 3 (2010), 607–625. https://doi.org/10.1007/s11276-008-0157-7

[6] Cristiano Castelfranchi. 2000. Founding agent's 'autonomy' on dependence theory. In *Proceedings of the 14th European Conference on Artificial Intelligence*. IOS Press, 353–357.

[7] Jiming Chen, Shijian Li, and Youxian Sun. 2007. Novel deployment schemes for mobile sensor networks. *Sensors* 7, 11 (2007), 2907–2919. http://www.scopus.com/inward/record.url?eid=2-s2.0-36849032166

[8] Daniel-Ioan Curiac. 2016. Towards wireless sensor, actuator and robot networks: Conceptual framework, challenges and perspectives. *Journal of Network and Computer Applications* 63 (mar 2016), 14–23. https://doi.org/10.1016/j.jnca.2016.01.013

[9] S. Damer, L. Ludwig, M. Anderson LaPoint, M. Gini, N. Papanikolopoulos, and J. Budenske. 2006. Dispersion and exploration algorithms for robots in unknown environments. *Unmanned Systems Technology VIII, SPIE Digital Library* (2006).

[10] Adam Dunkels. 2011. *The contikimac radio duty cycling protocol*. Swedish Institute of Computer Science.

[11] Konstantinos P. Ferentinos and Theodore A. Tsiligiridis. 2007. Adaptive design optimization of wireless sensor networks using genetic algorithms. *Computer Networks* 51, 4 (2007), 1031–1051. http://dblp.uni-trier.de/db/journals/cn/cn51.html#FerentinosT07

[12] Mirgita Frasheri, Baran Çürüklü, Mikael Ekström, and Alessandro Papadopoulos. 2018. Adaptive Autonomy in a Search and Rescue Scenario. In *12th IEEE International Conference on Self-Adaptive and Self-Organizing Systems*. 150–155. http://www.es.mdh.se/publications/5178-

[13] D. W. Gage. 1992. *Command control for many-robot systems*. Naval Command Control and Ocean Surveillance Center, RDT and DIV San Diego, CA.

[14] Luis Javier Garcia Villalba, Ana Lucila Sandoval Orozco, Alicia Triviño Cabrera, and Claudia Jacy Barenco Abbas. 2009. Routing Protocols in Wireless Sensor Networks. *Sensors* 9, 11 (2009), 8399. https://doi.org/10.3390/s91108399

[15] Brian Gerkey, Richard T Vaughan, and Andrew Howard. 2003. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th international conference on advanced robotics*, Vol. 1. 317–323.

[16] Amitabha Ghosh and Sajal K. Das. 2008. Review: Coverage and Connectivity Issues in Wireless Sensor Networks: A Survey. *Pervasive Mob. Comput.* 4, 3 (2008), 303–334. https://doi.org/10.1016/j.pmcj.2008.02.001

[17] Eva González-Parada, Jose Cano-García, Francisco Aguilera, Francisco Sandoval, and Cristina Urdiales. 2017. A Social Potential Fields Approach for Self-Deployment and Self-Healing in Hierarchical Mobile Wireless Sensor Networks. *Sensors* 17, 1 (2017), 120. https://doi.org/10.3390/s17010120

[18] Xiaoxiang Han. 2014. Mobile node deployment based on improved probability model and dynamic particle swarm algorithm. *Journal of Networks* 9, 1 (2014), 131–137. https://doi.org/10.4304/jnw.9.1.131-137

[19] Benjamin Hardin and Michael A Goodrich. 2009. On using mixed-initiative control: A perspective for managing large-scale robotic teams. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*. ACM, 165–172.

[20] Nojeong Heo and P. K. Varshney. 2005. Energy-efficient Deployment of Intelligent Mobile Sensor Networks. *Trans. Sys. Man Cyber. Part A* 35, 1 (jan 2005), 78–92. https://doi.org/10.1109/TSMCA.2004.838486

[21] A. Howard, M. Mataric, and G. Sukhatme. 2002. An incremental self-deployment algorithm for mobile sensor networks. (2002), 113–126 pages. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.4726

[22] EA Jensen and ML Gini. 2013. Rolling Dispersion for Robot Teams. *IJCAI* (2013). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.310.2769

[23] Nikitha Kukunuru, Babu Rao Thella, and Rajya Lakshmi Davuluri. 2010. Sensor deployment using particle swarm optimization. *International Journal of Engineering Science and Technology* 2, 1 (2010), 5395–5401.

[24] Raghavendra V. Kulkarni and Ganesh Kumar Venayagamoorthy. 2011. Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41, 2 (mar 2011), 262–267. https://doi.org/10.1109/TSMCC.2010.2054080

[25] Vlajic N. and Moniz N. 2007. Self-healing Wireless Sensor Networks: Results That May Surprise. In *Proc. of the IEEE Global Telecommunications Conference Workshops. GLOBECOM Workshops*. 333–338.

[26] Shaifull Nizam Othman. 2010. Node positioning in zigbee network using trilateration method based on the received signal strength indicator (\(RSSI\)). *European Journal of Scientific Research* 46, 1 (2010), 048–061.

[27] Recep Özdağ and Ali Karcı. 2016. Probabilistic dynamic distribution of wireless sensor networks with improved distribution method based on electromagnetism-like algorithm. *Measurement* 79 (feb 2016), 66–76. https://doi.org/10.1016/j.measurement.2015.09.056

[28] John H. Reif and Hongyan Wang. 1999. Social Potential Fields: A Distributed Behavioral Control for Autonomous Robots. (1999).

[29] Joshua Robinson, Eugene Ng, and Joshua Robinson. 2007. A Performance Study of Deployment Factors in Wireless Mesh Networks. In *in IEEE Infocom, 2007*. 2054–2062.

[30] Texas Instruments 2016. *CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver*. Texas Instruments. Rev. C.

[31] Simon Thiel, Dagmar Häbe, and Micha Block. 2009. Co-operative robot teams in a hospital environment. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, Vol. 2. IEEE, 843–847.

[32] Cristina Urdiales, Francisco Aguilera, Eva González-Parada, Jose Cano-García, and Francisco Sandoval. 2016. Rule-Based vs. Behavior-Based Self-Deployment for Mobile Wireless Sensor Networks. *Sensors* 16, 7 (2016), 1047. https://doi.org/10.3390/s16071047

[33] Lovekesh Vig and Julie A Adams. 2006. Multi-robot coalition formation. *IEEE transactions on robotics* 22, 4 (2006), 637–649.

[34] Lovekesh Vig and Julie A Adams. 2007. Coalition formation: From software agents to robots. *Journal of Intelligent and Robotic Systems* 50, 1 (2007), 85–118.

[35] Jin Yang, Fagui Liu, Jianneng Cao, and Liangming Wang. 2016. Discrete Particle Swarm Optimization Routing Protocol for Wireless Sensor Networks with Multiple Mobile Sinks. *Sensors* 16, 7 (2016), 1081. https://doi.org/10.3390/s16071081

[36] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. 2008. Wireless Sensor Network Survey. *Comput. Netw.* 52, 12 (2008), 2292–2330. https://doi.org/10.1016/j.comnet.2008.04.002

[37] Chun Zhang and Shumin Fei. 2013. Connectivity-Preserved and Force-Based Deployment Scheme for Mobile Sensor Network. *Wireless Personal Communications* 77, 1 (nov 2013), 463–475. https://doi.org/10.1007/s11277-013-1516-y