# An evaluation of two distributed deployment algorithms for Mobile Wireless Sensor Networks

Francisco Aguilera[1], Cristina Urdiales[1], and Francisco Sandoval[1]

ISIS group, ETSI Telecommunications, University of Malaga, 29071 Malaga, Spain
{cristina,sandoval}@dte.uma.es

**Abstract.** Deployment is important in large wireless sensor networks (WSN), specially because nodes may fall due to failure or battery issues. Mobile WSN cope with deployment and reconfiguration at the same time: nodes may move autonomously: i) to achieve a good area coverage; and ii) to distribute as homogeneously as possible. Optimal distribution is computationally expensive and implies high traffic load, so local, distributed approaches may be preferable. This paper presents an experimental evaluation of role-based and behavior based ones. Results show that the later are better, specially for a large number of nodes in areas with obstacles.

**Keywords:** Mesh mobile sensor networks, WSN, deployment, SPF, BDM

## 1 Introduction

Wireless sensor networks (WSN) are conformed by a set of spatially distributed sensors that connect to each other. Sensor nodes have some degree of autonomy and they include a processing unit, along with the sensor(s) and the communication chip. In many applications, networks have to be deployed over a large area in hard to predict configurations. Furthermore, a node battery life mostly depends on the amount of traffic that it is generating/routing. If a node falls, part of the network might get isolated. Furthermore, in WSN deployment has a strong impact not only in terms of coverage but also in connectivity and throughput [1]. WSN usually rely on a mesh topology, which is more redundant, but also more robust and easier to expand and modify. Unfortunately, it is also harder to set up and maintain.

Mobile WSN (MWSN) are a potential solution to this problem [2]. MWSN can modify their positions to reorganize the network on a need basis. This solution is particularly adequate for deployment in hazardous or remote areas, or in large areas involving a large number of nodes. MWSN may be deployed by multiple robots systems (MRS), so that each node can decide when and where to move itself.

This work focuses on creating and testing a robot swarm to deploy a mobile WSN as efficiently as possible in terms of coverage and energetic efficiency. The key idea under this concept is to use as little nodes as possible to cover a given area and to make it last as long as possible before batteries need to be replaced.

Robot nodes have been used to repair MWSN by replacing fallen nodes [3], but deployment typically requires robot coordination and localization. A common approach to localization is Received Signal Strength (RSS) based triangulation, since GPS sensors are battery consuming, heavy and expensive for this kind of application. In most approaches, a static beacon infrastructure is fixed for mobile nodes to locally position themselves [4] [5].

There are two main distributed deployment strategies. In centralized deployment algorithms, a master node gathers information about the whole network and makes all decisions. This process is computationally expensive and usually involves techniques like Genetic Algorithms [6] or, to distribute calculation, Particle Swarm Optimization [7]. However, these approaches usually required more complex hardware and involve a higher traffic load. Alternatively, distributed algorithms require less complex calculation and less intense communication, since no node needs to be particularly aware of the state of the rest. These algorithms consider that local dispersion leads naturally to global dispersion, so each node makes decisions according to local factors, e.g. RSS.

We are going to evaluate two different distributed approaches. These approaches are representative of rule based deployment and behavior based deployment, respectively. The first one is the Backbone Dispersion Algorithm (BDA) [8]. In BDA, nodes move randomly until they fulfill some termination conditions. Each node only needs to evaluate how many nodes it can connect to. Once a robot stops, it won't move again. This algorithm is simple and requires little communication among node. However, its constraints do not fit well with the requirements of e.g. hierarchical networks. In these cases, a behavior based deployment algorithm may be required. We are going to use a Social Potential Fields (SPF) algorithm to represent this second group. This algorithm was originally proposed for swarm robots [9] and can be easily extended to mobile WSN by using RSS as an additional source of data. These algorithms basically model a set of forces of attraction and repulsion depending on the local environment of the node to determine its emergent motion.

From this point on, we will refer to nodes as robots, where each robot includes the communication module, required sensors and a processing unit, plus onboard distance sensors to avoid collisions.

## 2 Deployment algorithms

BDA [8] depends on a set of deployment rules:

- At least two robots and one of them belongs to the backbone: the robot keeps moving and avoiding obstacles to spread the network as much as possible
- A single robot that belongs to the backbone: to prevent loss of connectivity to the network the robot stops until another robot gets nearby
- At least a robot, but no robot belonging to the backbone: the robot joins the backbone, stops and notifies its change of status to the rest of the network
- No robot: the robot moves backwards until it finds some robot to connect to

To implement the SPF [9], each robot is affected by two different forces:

– A repulsion force $f_{r1}$ that forces the robot away from other nearby robots or obstacles to prevent collisions.
– A second repulsive force $f_{r2}$ that aims at expanding the network.
– An attraction force (clustering force) $f_c$ that grows along with the distance between robots to prevent loss of communication.

Robots stop when these forces reach an equilibrium. To avoid the well known oscillation problem, an equilibrium threshold $f_u$ is used.

In BDA, no distance estimation is required [10]: robots are either connected or not. In SPF, however, we need to estimate distances to obstacles, including other robots within communication range. Each robot estimates these distances using the Friis equation [11] and RSS from nearby robots. Although this estimation is rough, it works correctly for reactive algorithms like BDA where no optimization is required. Distances allowed between robots mostly depend on our communication range and safety concerns, i.e. how close to obstacles we let the robot be. In our case we are going to operate with IEEE 802.14.4 standard and small robots. Hence, we use the following (heuristically estimated) parameters.

$$f_r 1(r) = -\frac{0.01}{r^8} \tag{1}$$

Using this adjustment, the repulsion equilibrium point between robots is equal to 1 meter. However, since our $f_u$ is equal to 0.5, this force only operates when robots are closer than 60 cm from each other. On the other side, the attractive force starts to be noticed when robots are at least 1.5 meters away:

$$f_c(r) = -\frac{20}{r^6} + \frac{2}{r^{0.2}} \tag{2}$$

Finally, the second repulsion force tries to keep a distance of approximately 2 meters between each two robots:

$$f_r 1(r) = -\frac{60}{r^7} \tag{3}$$

## 3 Evaluation parameters

In order to evaluate the results of deployment algorithms, we need to define a set of parameters of interest, first.

Coverage is used as a quality measure in networks. Specifically, we are going to evaluate blanket coverage: any point of the region is sensed by at least one sensor. If node $i$ covers a round area $A_i$, given $N$ sensors in a full area $A$, coverage $C$ can be calculated as:

$$C = \frac{\cup_{i=1 \ldots N} A_i}{A} \tag{4}$$

Eq. 4 can be modeled using a probabilistic grid of $M$ cells [12], where each cell $i$ yields the overall probability of detecting an event on that location $P_i$.

$$C = \sum_{i=1}^{M} \frac{P_i}{M} \tag{5}$$

Since events at cell $i$ can be detected independently by several nodes, $P_i$ needs to be calculated as follows:

$$P_i = 1 - \bar{P}_i = 1 - \prod_{N} (1 - P_{ij}) \tag{6}$$

$N$ being the number of nodes and $P_i j$ being the probability of node $j$ detecting an event at cell $i$.

Regarding energetic efficiency, we need to take into account two different energy costs: i) deployment and ii) maintenance. Deployment costs mostly depend on two parameters [13]: distance $d$ that each node covers to reach its final location; and time $t$ to reach the final location. After the deployment is complete, energy cost is usually related to how regularly nodes are distributed. Uniformity $U$ for $N$ nodes can be defined as:

$$U = \frac{1}{N} \sum_{i=1} U_i \tag{7}$$

$$U_i = \left(\frac{1}{K_i} \sum_{j=1}^{K_i} (D_{i,j} - M_i)^2\right)^{1/2} \tag{8}$$

$K_i, j$ being the number of nodes close to node $i$, $D_{i,j}$ being the distance between nodes $i$ and $j$ and $M_i$ being the average distance between node $i$ and its closest ones. The better $U$ is, the lower the network energy consumption.

There are many other factors that affect energy consumption after deployment, mostly related to routing strategies. In order to evaluate them indirectly, efficiency can be roughly estimated in terms of the average power that nodes require to send a message to the rest of the network $P_m$:

$$P_m = \frac{1}{N} \sum_{i=1}^{N} P_{mi} \tag{9}$$

$$P_{mi} = \frac{1}{N} \sum_{i=1}^{N-1} P_{ij} \tag{10}$$

$P_{mi}$ being the power required at node i to send a message to the rest of the network and $P_{ij}$ being the power required to send a message from node $i$ to node $j$. In networks where messages need to be retransmitted through $k$ nodes,

$$P_{ij} = P_{i1} + ... + P_{ik} \tag{11}$$

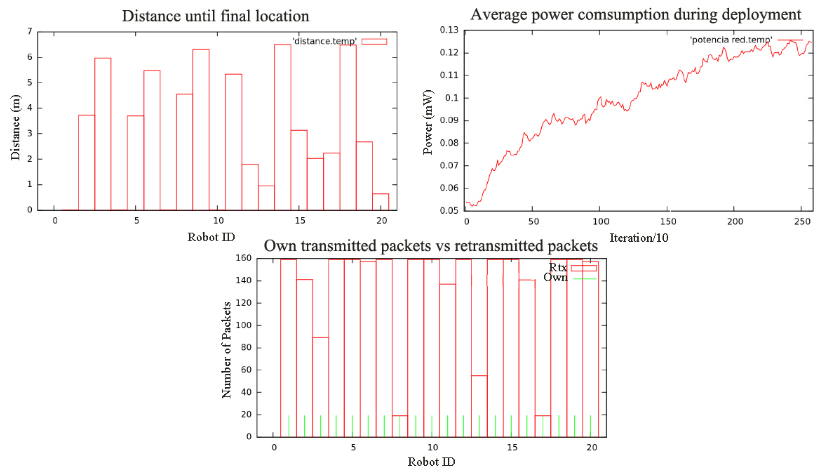


Fig. 1: BDA deployment for 20 robots in an environment without obstacles

# 4 Experiments and results

Although the following tests have been performed with a small number of physical robots (TI CC4305137) in real environments, in order to evaluate the impact of our different deployment strategies, we need a large number of robots. Hence, the following experiments have been performed using the freeware Player/Stage environment. Player allows us to control both a real and a simulated robot in an almost transparent way[14].

| N | Dply | Obst | t | d | C | U | $P_m$ | $Msg_{Tx}$ |
|---|---|---|---|---|---|---|---|---|
| 20 | BDA | N | 2m 3s | 2.140 | 79.33% | 0.578 | 0.052 | 34.8 % |
| 20 | BDA | Y | 3m 8s | 3.218 | 76.78 % | 0.574 | 0.051 | 32.5% |
| 100 | BDA | N | 6m 48s | 30.077 | 58.13 % | 0.533 | 0.119 | 34.7 % |
| 100 | BDA | Y | 7m 31s | 40.145 | 52.302 % | 0.524 | 0.113 | 33.36% |
| 20 | SPF | N | 1m 6s | 1.698 | 95.11 % | 0.706 | 0.071 | 32.7% |
| 20 | SPF | Y | 1m 46s | 2.360 | 99.33 % | 0.806 | 0.078 | 35.3% |
| 100 | SPF | N | 2m 38s | 18.499 | 93.67 % | 0.727 | 0.176 | 42.8% |
| 100 | SPF | Y | 2m 59s | 20.513 | 88.44 % | 0.706 | 0.166 | 42.7% |

Table 1: Deployment of a mesh network using BDA and SPF

In our mesh network simulations, nodes transmit at -10 dBm, corresponding to our measures using the real robots. We are going to focus uniquely on two simple routing mechanisms: flooding and closest neighbor. In order to avoid overflow in flooding, each robot can retransmit a message only once. In closest neighbor routing, a robot only retransmit a message if it is closer to the destination robot than the one it received the message from. There are much better routing techniques, but deployment can be evaluated simply with these two.

Table 1 shows simulations results for 20 and 100 robots using BDA for environments of 6 and 15 $m^2$, respectively, both without and with static obstacles.

We can observe that both average distance $d$ and deployment time $t$ grow slightly in presence of obstacles and largely with the number of robots in the group. However, coverage is actually poorer when a large number of robots is involved, even though the density of robots is larger in the second case. This fact outlines that distribution is subpar in BDA. Fig. 1 provides further detail at robot level for these scenarios. We can observe that $d$ changes significantly from one robot to another, since those joining the backbone first stop early during deployment. We can also observe that $P_m$ grows when they move in the 100 robots scenario, because they are more distant from each other. Finally, we can observe that, after deployment, the ratio of retransmitted packets for each robot with respect to own transmitted packets changes significantly from one robot to the next and, in most cases, is quite large. As a general rule, robots on the network boundaries retransmit less packages than the rest, although robots in very crowded areas also have a lower retransmission rate because nearby robots take part of the job.

Fig. 2 shows BDA deployment at two stages of a simulation with 100 robots in an environment without obstacles. We can observe in Fig. 2.a one of the main issues of this approach: during deployment, many robots get trapped in the center of the network. Although the problem eventualy fixes itself, deployment time gets considerably increased when this happens (particularly when the number of robots is large). In order to keep deployment time limited, we stop the simulation as soon as at least 80% of the swarm is deployed. Fig. 2.b shows the final location of the robots in this simulation. We can easily observe that there are uncovered areas and distribution is not homogeneous. An additional drawback of this approach is that robots tend to conform lines. These formations are particularly weak with respect to node failures, that might lead to disconnections of a mild number of nodes.

The presented results improve largely if we switch to SPF deployment. Table 1 shows results for 20 and 100 robots using SPF for environments of 6 and 15 $m^2$, respectively, both without and with static obstacles. We can easily observe that deployment is faster, robots move less, and both coverage and uniformity improve. It can also be observed that $P_m$ and $M\bar{s}g_{Tx}$ grow with the number of robots more than in BDA coverage. This happens because robots are better distributed and not so many of them are far from the rest or in very crowded areas. Fig. 3 shows the result of a 100 nodes network deployment in an environment

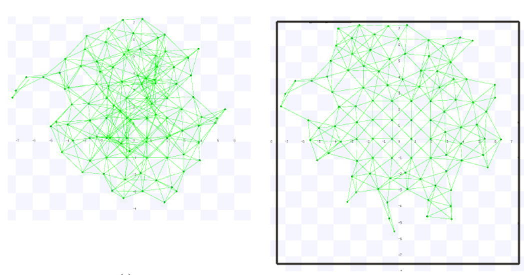


Fig. 2: BDA deployment: a) trapped robots; b) final configuration

without obstacles. All simulation in these environments return similar results. We can observe that nodes spread quite uniformly over the whole area.

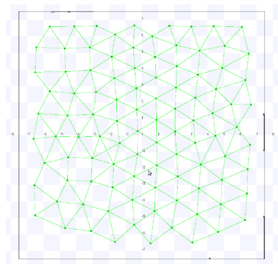

Fig. 3: SPF deployment for 100 robots in an environment without obstacles

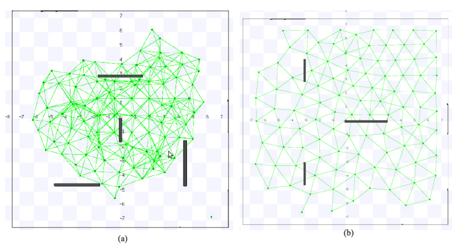

Fig. 4: Deployment for 100 robots in an environment with obstacles using: a) BDA; b) SPF

All commented results are more evident in environments with obstacles. Fig. 4 shows a sample environment with three obstacles. BDA is very affected by the location and shape of these obstacles, whereas SPF deployment basically adapts to the obstacles to obtain a configuration as similar as possible to Fig. 3.

## 5 Conclusions

We have presented an experimental evaluation of two representative distributed deployment algorithms for mobile WSN: rule based (BDA) and behavior based (SPF) ones. Nodes deploy faster and more homogeneously using SPF, plus they cover less distance in general. These effects are more acute the larger the network is. Average power and message retransmission is actually larger in SPF, because nodes are better distributed. In BDA, nodes on the boundaries and those in high density node areas do not have to retransmit so often. However, most loaded nodes fall quite early and, hence, full areas of the network get disconnected. In SPF, traffic load is better distributed, so the network lasts longer. SPF also adapts much better to areas with obstacles. In brief, SPF seems more suitable for deployment in large, dynamic, unstructured areas. Future work will focus on extending this study to hierarchical WSN.

## 6 Acknowledgements

This work has been partially supported by the Spanish Ministerio de Educacion y Ciencia (MEC), Project n. TEC2011-06734, by Junta de Andalucia, Project n. TIC 7839 and by International Campus of Excellence Andalucia Tech.

## References

1. Robinson, J., Ng, E., Robinson, J.: A performance study of deployment factors in wireless mesh networks. In: in IEEE Infocom, 2007. (2007) 2054–2062
2. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw. **52**(12) (August 2008) 2292–2330
3. Mei, Y., Xian, C., Das, S., Hu, Y.C., Lu, Y.H.: Repairing sensor networks using mobile robots. In: Proc. of the ICDCS International Workshop on Wireless Ad Hoc and Sensor Networks (IEEE WWASN 2006). (2006)
4. Batalin, M.A., Sukhatme, G.S.: Coverage, exploration and deployment by a mobile robot and communication network. In: Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks. (2003) 376–391
5. Hattori, K., Owada, N.T.T.K.Y., Hamaguchi, K.: Autonomous deployment algorithm for resilient mobile mesh networks. In: Proc. of Asia-Pacific Microwave Conference. (2014) 662–664
6. Ferentinos, K.P., Tsiligiridis, T.A.: Adaptive design optimization of wireless sensor networks using genetic algorithms. Computer Networks **51**(4) (2007) 1031–1051
7. Kukunuru, N., Thella, B.R., Davuluri, R.L.: Sensor deployment using particle swarm optimization. International Journal of Engineering Science and Technology **2**(1) (2010) 5395–5401
8. Damer, S., Ludwig, L., LaPoint, M.A., Gini, M., Papanikolopoulos, N., Budenske, J.: Dispersion and exploration algorithms for robots in unknown environments. Unmanned Systems Technology VIII, SPIE Digital Library (2006)
9. Reif, J.H., Wang, H.: Social potential fields: A distributed behavioral control for autonomous robots (1999)
10. Ludwig, L., Gini, M.: Robotic swarm dispersion using wireless intensity signals. Distributed Autonomous Robotic Systems **7** (2006) 135–144
11. Han, C., Yim, J., Lee, G.: A review of friis equation based indoor positioningn. Advanced Science and Technology Letters **30** (2013) 101–105
12. Ghosh, A., Das, S.K.: Review: Coverage and connectivity issues in wireless sensor networks: A survey. Pervasive Mob. Comput. **4**(3) (June 2008) 303–334
13. Heo, N., Varshney, P.K.: A distributed self spreading algorithm for mobile wireless sensor networks. IEEE Wireless Communication and Networking (2003) 1597–1602
14. Gerkey, B.P., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: In Proceedings of the 11th International Conference on Advanced Robotics. (2003) 317–323