Distributed Algorithm for Robotic Network Self-deployment in Indoor Environments Using Wireless Signal Strength

Renato Miyagusuku, Atsushi Yamashita, and Hajime Asama

Department of Precision Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Abstract. Mobile robots with wireless capabilities can enable network connectivity over large areas by retransmitting wireless signals from a ground station. Our goal is, for such robotic networks, to enhance current teleoperated robots' ability to perform reconnaissance or assist human first responders on victims' search and rescue operations. On these missions, uninterrupted communications between teleoperated robots and their human operators are essential. In order to maximize the teleoperated robots' working area, the robotic network has to deploy itself, in an unknown and potentially hazardous environment, spreading as much as possible without losing network connectivity. In this paper, we consider an urban search and rescue setting; and present a distributed algorithm that allows simple mobile robots to self-deploy and create robotic networks, without the need of advanced self-localization capabilities nor prior knowledge of the environment.

Keywords: Multi-agent systems, Networked robotics, Wireless signal strength

1 Introduction

The fathomable Tohoku earthquake and tsunami on 2011 caused (other than 3 reactors' meltdowns at Fukushima nuclear power plant) almost 130 thousand buildings to collapse, over 250 thousand to semi-collapse, and partially damaged almost 700 thousand; moreover, it claimed almost 16 thousand lives. After such natural disaster, survivors may be trapped inside damaged buildings. Search and rescue operations play a crucial role in saving these lives; unfortunately, human first responders are often impeded from effectively searching throughout all building areas due to crumbling walls, poor visibility, narrow passages or simply lack of man power. An important application of robotic systems would be the assistance at urban search and rescue operations in such disaster-stricken scenarios as robots can access unsafe locations, support victim search and aid on victim rescue operations. Although its importance, current rescue robotic systems fall short to accomplish such tasks, being limited range/unreliable communications a mayor current drawback [16]. Maintaining such communications would be a trivial matter if fair wireless signals were available at any point within the affected area; however, this hardly ever holds true on real applications, especially on disaster-stricken scenarios where wired networks most likely malfunction and collapsed walls, obstacles and debris may hinder wireless signals. Notably, these collapsed walls, obstacles and debris can substantially modify the environment, making available maps or blue prints inaccurate enough as to render map-based localization not a viable option; therefore it's fair to assume these collapsed, semi-collapsed or damaged buildings as unknown environments, even if previous maps or blueprints are available.

The work herein presented intends to solve this problematic situation by developing a multi-agent system composed by several mobile robots with wireless capabilities - robotic routers. Each of these robots can expand wireless area coverage by associating with an ad-hoc network and retransmitting its signals; enabling communications between a ground station and any mobile user (teleoperated or autonomous robot) within the area. In order to make the system robust, the robots should place themselves as to create a bi-connected graph (robots as nodes and available communication links as graph edges); if this is achieved, in case of single-robot failure, no matter which robot fails, the entire network would not collapse. Figure 1 illustrates our intended system, and its optimal topology; with robotic routers R1 to R6 self-deploying in a bi-connected graph that expands the network's area coverage.

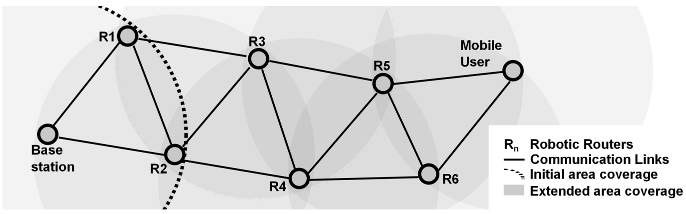


Fig. 1: Robotic routers 1 to 6 (R1 to R6) expand Base station's network coverage by relaying the messages transmitted from Base station to the Mobile User.

To attain a scalable system we pay attention to algorithm complexity with respect to the number of robots. Additionally, as uninterrupted communications are prioritized, we prefer algorithms with minimum to none inter-robot communications as to save main communication bandwidth for commands and/or video streaming. Regarding system inputs, we limit robots to use only locally available data, relaying on robot's previous states, wireless signal strength measurements, odometry and basic proximity sensors. Given these considerations, we chose a behavior-based approach which efficiently allows them to self-deploy maximizing network coverage, without compromising network connectivity. However, although efficient and scalable, our algorithm does not provide connectivity guarantees with mobile users (rescue robots), nor explicitly guarantees graph bi-connectivity, both issues remaining for future work.

2 Related Work

Network deployment and maintenance is a topic of great interest given its vast amount of applications: search and rescue [5], target detection [12, 14], hazardous contamination [13], large area search [1], among others. In [15] fixed network wireless repeaters are placed, moved and even repaired by a mobile robot, as to establish and maintain a wireless sensor network; this approach, although interesting, is not applicable to our current problem, as having a dynamic environment with continuously varying requirements due to mobile users' actions, makes the use of fixed nodes not feasible. Another approach is to have a group of robots traveling behind the user and becoming repeaters when required [11], having mobile robots gives flexibility to the network, the ability to easily fall back after mission is completed, and, more importantly, it guarantees the network to be fully connected at any point of time. A similar approach is taken in [7], but considering the use of an hybrid (wired/wireless) system, where when feasible robots transport a wired connection, and up to a certain distance, they start relaying on wireless repeaters. However, as both approaches en up creating a single chain of robots, it makes the system quite vulnerable to single-robot failure, as it would cause the whole network to collapse.

Another well studied field is robot dispersion in indoor environments. Studies like [3, 10, 8] present several distributed algorithms for dispersion of groups of autonomous mobile robots in indoor environments; all of which present core ideas and behaviors similar to ours. However, they all keep line of sight between robots, as they use infrared sensors, sonar arrays or laser range finders to obtain distance and bearing information from its neighbors - some also use prior knowledge of the map or explicit inter-robot communication. Our approach widely differs from these studies as we do not require keeping line of sight. Each of our robots employs its neighbors' wireless signal strength information as main measurements for estimating their relative positions.

Our approach utilizes signal strength as main sensory input. Therefore, it is important to point out that there is debate whether signal strength measurements are a reliable distance estimator, works like [4] and [6] consider it as a poor estimator, with [6] placing distance estimation errors between 3 and 9 m. However, both works consider static access points, while ours are mobile; this mobility can be exploited to decrease estimation errors. Furthermore, our algorithms are based on signal strength measurements and bearing estimation to closest neighboring robots, not distance estimation. Bearing estimations can be derived from Particle Filters using noisy, inaccurate signal strength measurements (as the ones mentioned at [6]) and simple odometry information, as shown in our previous work [9].

3 Network Deployment

It is assumed that robotic routers are equipped with proximity sensors for obstacle avoidance, as well as basic odometry sensors. It is also assumed they have 802.11-compliant wireless network interfaces. Standard 802.11 cards have a built-in received signal strength indicator (RSSI) that will be used by the robots to acquire signal intensities. Using odometry and these RSSI values, we assume bearings to closest neighboring robots can be estimated. Under these assumptions, we develop a behavior-based system that can effectively spread robotic routers across a delimited area, without losing connectivity.

This swarm-like approach considers simple robots, all executing the same software program that only states local decisions; however at the system-level all these decisions will produce the intended emergent behavior. The algorithm uses simple local rules, called behaviors; which are elements of a behavior-based architecture. This architecture decides which behavior should command the agent at any given moment, given the current inputs and robot's past state.

3.1 Behaviors

In our approach, the behaviors we employ are: *collision-avoidance, random-walk, dispersion, aggregation* and *wall-following-exploration.*

Collision-avoidance behavior is straight forward: *If an obstacle is detected too close, get away from it.* This obstacle can be a wall, debris or malfunctioning robots. Inputs for the collision-avoidance behavior are proximity sensors, such as sonars or infra-red sensors. This behavior has the highest priority and can over-rule any other primitive.

Random-walk behavior is another quite intuitive behavior: *If neighbors are too close, choose a random direction and move away towards that direction.* The algorithms employs signal strength measurements as inputs; and its purpose is twofold: to try regaining connectivity if the robot completely loses connection to all its neighbors; and to have an effective way to spread the network if robots are so close together that estimation errors in distance and bearing make other behaviors unreliable.

Dispersion behavior's goal is to spread the network; the basic idea being *If close, spread away from neighboring robotic routers.* There are many approaches as how to compute this behavior. In our case, our main interest is not spatial separation of robots, but rather wireless network coverage; therefore, signal strength is used rather than distance. The behavior takes as an input a desired signal strength value, which is used as a threshold - dispersion threshold S_d. When a neighboring robotic router's signal strength is higher than S_d, meaning it is too close, the dispersion behavior calculates a virtual force F_d that moves the robot towards its neighbor's opposite direction. Equation (1) shows how this virtual force is calculated:

$$F_d = -m_d(S - S_d)\hat{\mathbf{u}}, \tag{1}$$

where S is the signal strength, m_d is a positive constant and $\hat{\mathbf{u}}$ is the robotic router's bearing to its neighbor. These forces are generated considering only the c closest neighbors that comply with the condition; and the final output of the behavior is the sum of all these forces. This c parameter is set by the designer and its effects over the network are addressed in section 4.

Aggregation behavior acts oppositely to the dispersion behavior, in the sense that its goal is for the network not to spread too much. Its main principle is *If a neighboring robotic router is so far that you could lose connectivity, move towards it*, this as to avoid loss of connectivity. Like the dispersion behavior it computes virtual forces. If the signal strength measured from a neighboring robot is lower than a second threshold (aggregation threshold S_a), the aggregation behavior calculates a virtual force F_a that moves the robot towards its neighbor. Equation (2) shows how this virtual force is calculated:

$$F_a = m_a(S_a - S)\hat{\mathbf{u}}, \tag{2}$$

where m_a is a positive constant. Like the previous behavior, these forces are generated only with the c closest neighbors that comply with the condition; and the output is the sum of the forces generated.

Wall-following-exploration behavior is used to explore the map by following the first wall the robots bumps into. The robot follows the wall by turning a random angle and moving forward with a small curvature until it bumps/detects the wall; once it does, it turns away a small angle and starts moving forward with the same small curvature again until it bumps/detects the wall once more; this behavior is repeated several times. This behavior is crucial for dispersion on structured environments like offices, where robots only ruled by the previous behaviors often get stuck and are not able to leave the room they are in and expand the network.

3.2 System Architecture

We employ an architecture similar to ALLIANCE [13]; our approach differs in not considering inter-robot communications, as the algorithm doesn't require them, and in only using a slight modification of *acquiescence* as internal motivation, instead of *acquiescence* and *impatience*. Under this framework we group behaviors in three sets (BS0, BS1 and BS2). Each behavior set has a different emergent (high-level) behavior and a goal; these different goals may contradict each other, so only one of the behavior sets can be active at any point in time. The behavior sets' outputs are activated by their respective motivational behaviors, which cross-inhibit each other - as to guarantee only one of them is active. One of our behavior sets' activation is directly inhibited by the other two, as they may have higher priority; while the other two inhibit each other by means of the internal motivation *acquiescence*. Figure 2 shows a diagram of our system's architecture.

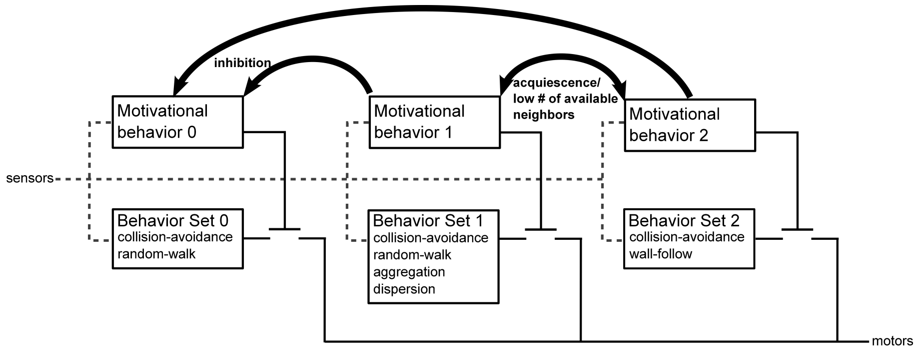


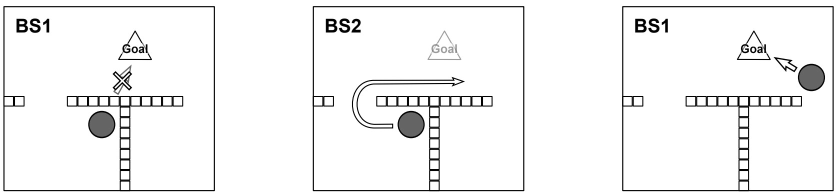
Fig. 2: System Architecture implemented on each robot. The system has three behavior sets that have different goals and perform different emergent behaviors.

Behavior Set 0 has the collision-avoidance and the random-walk behaviors. It is activated when the robot does not detect any neighboring robot; and its emergent behavior is safe wandering. BS0's purpose is to allow the robot to wander around the map until it finds other robots with which establish a network. This is an emergency behavior set, which should usually remain inactive as ideally the robots do not lose complete connectivity from their neighbors at any given point in time. The activation of either BS1 or BS2, inhibits BS0.

Behavior Set 1 has the collision-avoidance, random-walk, aggregation and dispersion behaviors. Only one of these internal behaviors is active at any point of time, and the selection is based on hierarchy, with the collision-avoidance behavior having the highest priority and the dispersion behavior the lowest. Collision-avoidance is used for safe navigation; following in priority, is the random-walk behavior which activates if the robots are so close together that errors in measurements and/or bearing estimations make the aggregation and dispersion behaviors unreliable. The aggregation behavior advocates for network connectivity while dispersion behavior for area coverage; as connectivity is more important than area coverage, we set aggregation with priority over dispersion. BS1's emergent behavior is safe network dispersion, allowing the system to efficiently self-deploy on obstacle free environments. However, on highly structured environments, such as offices, this behavior set alone is insufficient for an adequate deployment - as it will be shown on section 4.2. The behavior set is activated by default (if neither BS0 nor BS2 are active) or if the number of available neighbors is low, in which case the set is activated so the aggregation behavior can maintain connectivity.

Behavior Set 2 has the collision-avoidance and wall-following-exploration behaviors. Its emergent behavior is safe environment exploration. The set's purpose is to free robots that get stuck trying to move when it is physically impossible for them to do so - like moving forward when there is a wall in front; BS2 allows such robots to break free from this impossible task, and expand the network by exploring another area. BS2 is activated when internal motivation *acquiescence* goes over a fixed threshold, and remains active until it "runs out"; this *acquiescence* builds up when it encounters this impossible tasks.

Acquiescence is the only internal motivation used in our approach. Figure 3 illustrates our proposed *acquiescence* motivation; like ALLIANCE's *acquiescence* [13], we consider ours as a value that increases as the robot tries to perform a task, but sensory feedback indicates it is not succeeding in doing so - essentially trying to position itself somewhere but failing (Fig. 3a). Once *acquiescence*'s value reaches a threshold it inhibits BS1 and activates BS2 - and its wall-following-exploration behavior. While BS2 is activated *acquiescence*'s value starts dropping with time (Fig. 3b). Once it reaches zero, BS1 is activated and BS2 inhibited (Fig. 3c); importantly, not to neglect network connectivity, *acquiescence*'s value can also be forced to 0 if the number of available neighbors becomes low - so the aggregation behavior in BS1 is activated and network connectivity can be maintained.



(a) *Acquiescence*↑ (b) *Acquiescence*↓ (c) *Acquiescence* = 0

Fig. 3: Internal motivation: *Acquiescence*

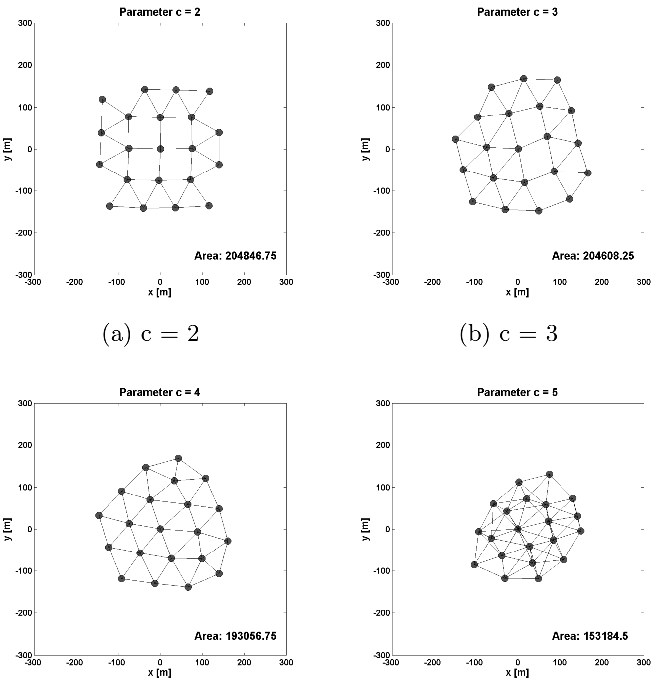
4 Simulations and Results

We analyze our system in two types of indoor environments: an open, obstacle-free; and a structured, office-like, environment. The obstacle-free environment is unlikely to be found in practical applications; however it is useful to study some of the algorithm's properties.

4.1 Obstacle-free Environment

When self-deploying in an obstacle-free environment, BS2 is never activated, as nothing hinders robots' movements. Therefore, on normal operation, only BS1 becomes active. Figure 4 shows different examples of BS1 when deploying 21 agents (gray circles), 20 acting as robotic routers and one, being anchored at (0,0), acting as a ground station. For simulations on obstacle-free environments we consider a maximum signal radius for communications equivalent to 100 m - over this distance communications are considered unreliable and connectivity is broken, S_a of 80 m and S_d of 75 m. At starting configuration all routers are clustered together and have an effective wireless coverage area of 31416 m^2; Fig. 4a shows the final configuration with parameter c =2, which has an area coverage of 204846.75 m^2 (6.5X initial area).

This before mentioned parameter c - number of closest neighbors considered for forces computation affects how interconnected each agent remains with respect to its neighbors, effectively increasing the systems redundancy to communication link lose but decreasing the area coverage. Some examples of configurations with different c values are shown at Fig. 4. Interestingly from the network configuration (with parameters c = 2, 3 and 4) it can be seen that even though no explicit behavior regarding the shape of the network was given, it has naturally formed a triangle based grid; notably, this triangular configuration is the most appropriate to guarantee network connectivity while maximizing area coverage, and was the one to illustrate our intended system on Fig 1.



(a) c = 2 (b) c = 3

(c) c = 4 (d) c = 5

Fig. 4: Different network configurations obtained after the deployment of 20 robotic routers and a ground station for different c values

After performing several simulations with different c parameters (2, 3, 4 and 5), the mean and standard deviation of the system's area coverage and robot's minimum number of connections were found. The area coverage for c =2 and 3 are quite similar, while for $c = 4$ is 5.8% smaller, and for $c = 5$ is 19%. Robot's minimum number of connections is an indication of the systems redundancy, the more connections each agent possess the more redundant the system is to communication link lose. If each robot possesses at least 2 connections, then even if one connection is severed the network connectivity remains intact; this redundancy is the one we desire. The mean of the robot's minimum number of connections for $c = 2$ is 1.8±0.41, for $c = 3$ is 2.18±0.82, for $c = 4$ is 2.68±0.55 and, for $c = 5$ is 2.98±0.77; making for this particular implementation parameter $c = 4$ an appropriate choice, as it's a compromise between area coverage and minimum desired number of connections.

4.2 Office-like Environments

Now we analyze the performance of our system in office-like environments; for practical applications, especially for search and rescue operations, this kind of environments are the most likely to be found. For simulations we consider much more restricted signal strength thresholds, as we expect obstacles and debris to heavily hinder signal propagation - signal strength parameters were set to S_a equivalent to 16 m and S_d to 15 m. The scenario used for simulations is shown at Fig. 5, it is a 100x30 m^2 area with several rooms, doors and one main corridor. All robots (blue circles) are initially clustered in one room, where the ground station (black circle) is also located.

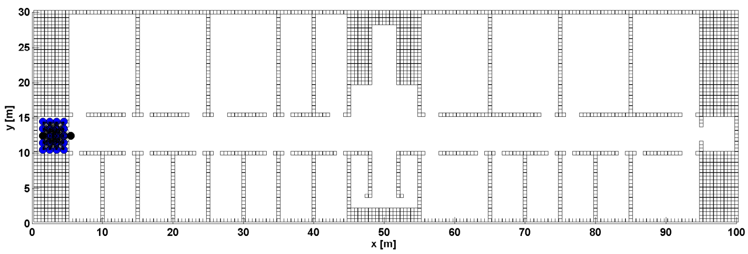
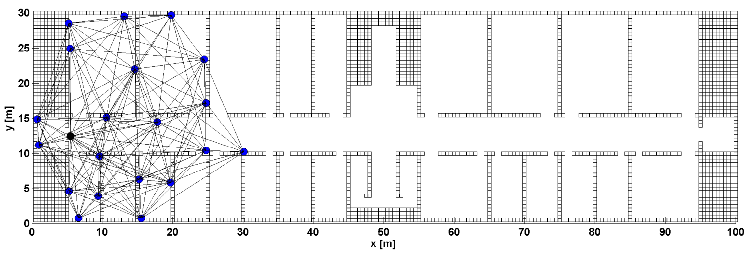


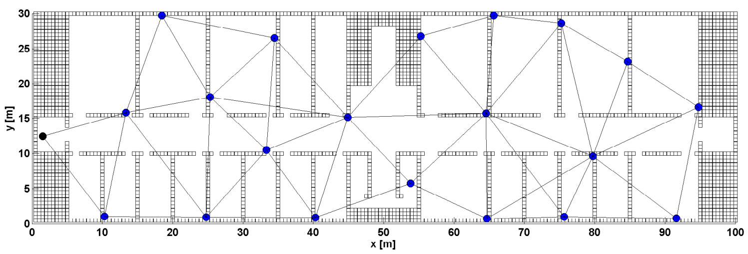
Fig. 5: Office-like scenario used for testing. 20 robotic routers and a ground station are initially placed clustered together close to the building's entry.

First we consider the same case as in the obstacle-free environment, with only BS1 activated - Fig. 6a shows the results of this approach. As it can be seen, robots fail to successfully position themselves on the map as to maximize area coverage; in fact, most of them remain in few rooms close to the initial position of deployment. This mainly occurs because robots get stuck due to walls positioned in the middle of their desired paths; as no exploration algorithm nor path planner has been implemented on the robots, they have no idea how to overcome these obstacles - the collision-avoidance behavior only allows robots to circumvent small obstacles, when faced with walls in a room, robots do not have any mean to overcome this.

However, once BS2 is introduced, area coverage notably improves; as it can be seen on Fig. 6b. As mentioned before, when a robot gets stuck due to a wall, *acquiescence* builds up, and after certain amount of time BS2 is activated. BS2 then forces the robot to give up on its current desired position, and instead makes it start exploring the environment by following the wall that previously restricted its movements. Once the behavior set is deactivated due to time, the robot goes back to BS1 and tries to find an adequate position abiding BS1 rules. The robot will perform these actions until it finds a suitable location, which improves the positioning of the robots all over the environment; however, as the exploration is random, network deployment requires a considerable amount of additional time - around 3 times that of deployment without BS2.



(a) System's self-deployment with only BS1. Robots get stuck in rooms near the entry, failing to further increase the network coverage to the whole map.



(b) System's self-deployment with BS2. By BS2, robots no longer get stuck, successfully covering the whole map. However this comes with an elevated required time for deployment and considerable energy spent moving.

Fig. 6: Effect of BS2 on the deployment of 20 robotic routers and a ground station in an indoor environment.

After running several simulations we found that on average, the area coverage, without BS2 activation, for our 3000 m^2 example scenario was 1294$\pm$103 m^2, 43.13$\pm$3.43% of the total area; and, with BS2 activation, was 2932$\pm$63 m^2, 97.73$\pm$2.1% of the total area. Demonstrating the importance of BS2 for indoor exploration in structured, office-like, environments.

5 Conclusions and future work

In this work we have demonstrated, using simulations, the feasibility of deploying a robotic network in indoor environments using only simple behaviors and no explicit inter-robot communications. This was achieved employing three behavior sets (BS0, BS1 and BS2). BS0 importance comes when under unexpected situations, the robot becomes totally disconnected from its neighbors. As the system is thought for disaster scenarios, these situations are expected. Specially on physical implementations, we believe BS0 will be of great importance. A more in depth analysis of such situations remains for future work.

Regarding BS1, its robustness against single-robot failure was addressed, and although it can be concluded that certain assurance of this robustness is in fact obtained just by varying parameter c, it still does not explicitly guarantee it. Work done by Ahmadi and Stone [2] may give a light into how to guarantee connectivity after single-robot failures; furthermore, if fast re-arrangement of the network into a bi-connected graph after single-robot failure can dote the system with a high degree of robustness even after multiple-robot failures.

The importance of BS2 for the network's deployment in office-like environment was established by simulation results, which showed that on average, for our particular study case, the network was able to cover more than twice the area than without it. These results can be generalized for any office-like environment with similar characteristics (abundance of rooms or offices). It is important to notice that network deployment in this kind of environments requires a significant amount of time and spends considerably more energy than in obstacle-free environments. However, as robots start exploring the area by following walls, it could be possible for them to store information from its sensors and use it to map the area, this information could be then sent to the base station and used to create a virtual model of the environment; with this information the system could start to use more complex strategies for exploration and, perhaps, significantly reduce the amount of time spent on deployment. All these ideas are yet to been implemented and remain as future work.

Finally, we would like to emphasize that, although simulations have shown promising results for our approach, we believe physical implementation and testing of the proposed algorithms in robotic platforms are key remaining issues of the work herein presented. Thus, we are actively working towards this.

References

1. Fabrizio Abrate, Basilio Bona, Marina Indri, Stefano Rosa, and Federico Tibaldi. Multi-robot map updating in dynamic environments. In *Distributed Autonomous*

Robotic Systems, pages 147–160. Springer, 2013.
2. Mazda Ahmadi and Peter Stone. Keeping in touch: Maintaining biconnected structure by homogeneous robots. In *Proceedings of the National Conference on Artificial Inteligence*, volume 21, page 580. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
3. Maxim A Batalin and Gaurav S Sukhatme. Spreading out: A local approach to multi-robot coverage. In *Distributed Autonomous Robotic Systems 5*, pages 373–382. Springer, 2002.
4. Karl Benkic, Marko Malajner, P Planinsic, and Z Cucej. Using rssi value for distance estimation in wireless sensor networks based on zigbee. In *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pages 303–306. IEEE, 2008.
5. Subhrajit Bhattacharya, Nathan Michael, and Vijay Kumar. Distributed coverage and exploration in unknown non-convex environments. In *Distributed Autonomous Robotic Systems*, pages 61–75. Springer, 2013.
6. Eiman Elnahrawy, Xiaoyan Li, and Richard P Martin. The limits of localization using signal strength: A comparative study. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 406–414. IEEE, 2004.
7. Koichi Gyoda, Yasushi Hada, and Osamu Takizawa. Performance analysis of the network models for the search robot rescue system in the closed spaces. In *IEEE International Workshop on Safety, Security and Rescue Robotics, 2007. SSRR 2007.*, pages 1–6. IEEE, 2007.
8. James McLurkin and Jennifer Smith. Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In *Distributed Autonomous Robotic Systems 6*, pages 399–408. Springer, 2007.
9. Renato Miyagusuku, Atsushi Yamashita, and Hajime Asama. Distributed algorithm for robotic network self-deployment using wireless signal strength measurements. In *Proceedings of the Japan Society for Precision Engineering Spring Conference*, 2014.
10. Ryan Morlok and Maria Gini. Dispersing robots in an unknown environment. In *Distributed Autonomous Robotic Systems 6*, pages 253–262. Springer, 2007.
11. Alejandro R Mosteo, Luis Montano, and Michail G Lagoudakis. Guaranteed-performance multi-robot routing under limited communication range. In *Distributed Autonomous Robotic Systems 8*, pages 491–502. Springer, 2009.
12. Chia Ching Ooi and Christian Schindelhauer. Energy-efficient distributed target tracking using wireless relay robots. In *Distributed Autonomous Robotic Systems 8*, pages 39–50. Springer, 2009.
13. Lynne E Parker. Alliance: An architecture for fault tolerant multirobot cooperation. *Robotics and Automation, IEEE Transactions on*, 14(2):220–240, 1998.
14. Lynne E Parker. Distributed algorithms for multi-robot observation of multiple moving targets. *Autonomous robots*, 12(3):231–255, 2002.
15. Tsuyoshi Suzuki, Ryuji Sugizaki, Kuniaki Kawabata, Yasushi Hada, and Yoshito Tobe. Deployment and management of wireless sensor network using mobile robots for gathering environmental information. In *Distributed Autonomous Robotic Systems 8*, pages 63–72. Springer, 2009.
16. Satoshi Tadokoro. *Rescue Robotics: DDT Project on Robots and Systems for Urban Search and Rescue*. Springer, 2009.