Ad-hoc wireless network coverage with networked robots that cannot localize

2009 IEEE International Conference on Robotics and Automation
Kobe International Conference Center
Kobe, Japan, May 12-17, 2009

Ad-hoc Wireless Network Coverage with Networked Robots that Cannot Localize

Nikolaus Correll, Jonathan Bachrach, Daniel Vickery and Daniela Rus

***Abstract*— We study a fully distributed, reactive algorithm for deployment and maintenance of a mobile communication backbone that provides an area around a network gateway with wireless network access for higher-level agents. Possible applications of such a network are distributed sensor networks as well as communication support for disaster or military operations. The algorithm has minimalist requirements on the individual robotic node and does not require any localization. This makes the proposed solution suitable for deployment of large numbers of comparably cheap mobile communication nodes and as a backup solution for more capable systems in GPS-denied environments. Robots keep exploring the configuration space by random walk and stop only if their current location satisfies user-specified constraints on connectivity (number of neighbors). Resulting deployments are robust and convergence is analyzed using both kinematic simulation with a simplified collision and communication model as well as a probabilistic macroscopic model. The approach is validated on a team of 9 iRobot Create robots carrying wireless access points in an indoor environment.**

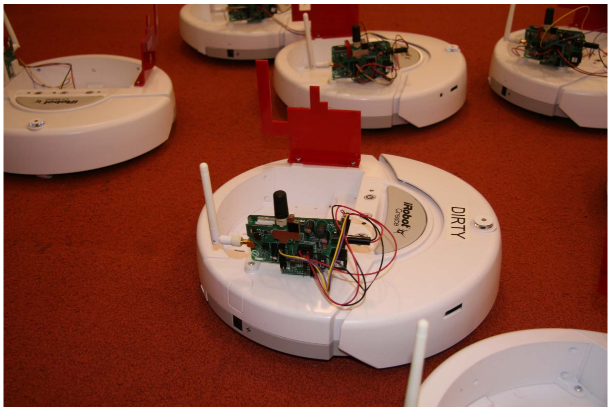

Fig. 1. A team of iRobot Create robots equipped with IEEE 802.11b wireless nodes running MIT Proto is deployed to create an ad-hoc network that covers an area with network access.

I. Introduction

We wish to deploy inexpensive robots with minimalist capabilities over an area and use the group as a computation, communication, and sensing backbone. The swarm will disperse autonomously to create a mobile communication network with maximum coverage. In this paper, we study this problem when the swarm of robots is minimalist in their resources and information — that is, the robots operate by using wireless connectivity only to estimate the network topology, have no means of localization, and avoid obstacles using bumper sensors. Applications include first response operations in areas where the existing computation and communication infrastructure has been destroyed, and monitoring remote areas with no infrastructure.

The robots' information and processing capabilities will affect the quality of the communication network that they can establish. The critical resources are (1) knowledge about the position of other robots (relative range-and bearing), (2) knowledge about the placement of the robot within the environment (global localization), and (3) knowledge about the environment in terms of maps. We see a hierarchy of capabilities, which ranges from no map, no localization, and local communication to full information on the environment, global localization, and complete knowledge of the position of each member in the swarm.

Our long-term goal is to precisely quantify the performance benefits as we move across this hierarchy toward increasingly more capable robots. In this paper, we focus on studying the lowest level of the hierarchy, that is a group of robots that can move in space, avoids obstacles, and can estimate the topology of the network within the range of its wireless communication device. While such a minimalist approach offers the advantage to allow for deployment of large numbers of small and cheap units, building up a system from the bottom-up will also help us to construct more robust algorithms that deal better with sensor and actuator noise present in more capable systems. More generally, designing a minimalist algorithm helps us to understand better the information structure of a robot task, that is the information and ressources which are necessarily needed by a robot system to accomplish a task.

A. Contribution of this paper

We develop a minimalist distributed algorithm that provides a scalable and robust solution to maximizing the coverage area of a wireless mesh network solely by relying on the number of other robots within communication range. We compare its performance to a hypothetical optimal solution. Convergence of the algorithm as a function of different parameter choices is analyzed using kinematic simulation and a probabilistic model. The results presented in this paper establish a baseline for the coverage performance that can be achieved with a robotic platform in its most

N. Correll, D. Vickery, and D. Rus are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02478, USA {nikolaus|rus}@csail.mit.edu,drv@mit.edu

J. Bachrach is with Makani Power Systems Inc. Alameda, CA 94501, USA jrb@pobox.com

This work was supported in part by the Swiss NSF under contract number PBEL2118737, MURI SWARMS project W911NF-05-1-0219, NSF IIS-0426838 and EFRI 0735953 Intel and Boeing. We are grateful for this support.

978-1-4244-2789-5/09/$25.00 ©2009 IEEE

basic configuration, and will help us to analyze benefits and costs of more sophisticated algorithms requiring additional capabilities. Finally, we present the development of a low-cost, mobile networking robotic platform running MIT Proto, an open-source programming language and virtual machine aimed at swarm robotic applications, which is used for validating the proposed algorithm in an indoor mesh network deployment scenario.

B. Related work

Algorithms for deploying large teams of agents can be well classified using the requirements they pose on the individual robotic nodes. For instance, some algorithms require noise-less, global localization and reliable, global information exchange among the nodes. Assuming perfect global localization and reliable information exchange allows for leveraging tools from graph and control theory for controller analysis and design (see also the special issue on networked control systems [1]). Particularly noteworthy with respect to the case study considered in this paper are distributed control approaches which exploit global metrics of graph connectivity for deriving local control laws [2].

Another class of algorithms are fully decentralized and reduce the requirements on the individual robotic node to provide the *relative* positions of its immediate neighbors and local information exchange [3]–[6].

Assumptions on the individual robotic platform are further relaxed by approaches that solely rely on the signal strength information between nodes [7], [8]. Only few contributions have adressed the network coverage problem using minimalist robotic nodes. For instance, [9] proposes an algorithm and analysis for coalescence of a team of robots to a static gateway based on random walk. As robots never resume motion once they stopped close to the gateway, this work is a special case of the algorithm and analysis presented in this paper.

II. Experimental Setup

We study the proposed algorithm at four levels of abstraction: a hardware implemention (Figure 1), a dynamical model implemented in the Proto simulator (Figure 2), a kinematic model implemented in Matlab, and an analytical probabilistic model. Experiments are conducted in the GPS-denied ground floor of the MIT Stata Center.

A. Hardware

For computation and networking we use an Atheros radio-on-a-chip MIPS platform running OpenWRT[1] Linux (180MHz, 32MB RAM, 8MB Flash). The transmission strength of the radio interface has been artificially limited to 5 dBm. For locomotion we are using a iRobot Create differential wheel robot (diameter around 30cm) that provides coarse odometry, bumpers and cliff sensors. The MIPS CPU communicates with the Create via a serial interface in intervals of 50ms. The current retail cost of the overall system, including NiMH battery and GPS is below $200 and has potential for further miniaturization.

B. Proto

MIT Proto[2] [10] is an open-source language and toolkit that makes it easy to write complex programs for spatial computers. A spatial computer is a collection of devices distributed to fill space, where the difficulty of communicating between devices is strongly dependent on their geometric distance. Examples include sensor networks, robotic swarms, cells during morphogenesis, FPGAs, ad-hoc wireless systems, biofilms, and distributed control systems. Proto is a language we have developed for programming spatial computers using a continuous space abstraction. Rather than to describe the behavior of individual devices, the programmer views the space filled by the devices as an amorphous medium—a region of continuous space with a computing device at every point—and describes the behavior of regions of space. These programs are automatically transformed into local actions that are executed approximately by the actual network of devices. When the program obeys the abstraction, these local actions reliably produce an approximation of the desired aggregate behavior. MIT Proto is our implementation of Proto, comprising a compiler, cross-platform virtual machine including a simulator, code libraries, and tutorial material. A screenshot of the Proto simulator is shown in Figure 2.

For this paper, we implemented a virtual machine on the MIPS platform that is able to execute Proto bytecode. Common to all platforms is the implementation of primitives for sending and receiving raw packets over the (platform specific) radio link. On a robotic platform, Proto requires additional opcodes for driving the robot given a vector as well as reading its sensor values that are then available as Proto expressions. The Proto simulator is a virtual machine running on Linux and implements the most commonly used sensors and actuators used across the supported platforms.

C. Networking

The Atheros wireless card is configured to provide two virtual devices that both operate on the same physical channel. One device is used by Proto in monitor mode to exchange connectivity information, including the number of hops to the gateway, with its neighbors. A second device is configured in ad-hoc mode and associates with neighboring nodes. This device is used by an implementation of the OLSR algorithm (OLSRd[3]) to provide TCP/IP routing for the resulting mesh network.

III. Problem Statement and Performance Metrics

We are interested in a network that both maximizes the area covered by the wireless signal as well as being sufficiently dense in order to be robust to unreliable links and failure of individual nodes, while being connected to a static gateway node through a finite number of hops. We are

[1] http://www.openwrt.org

[2] http://groups.csail.mit.edu/stpg/proto.html

[3] http://www.olsr.org

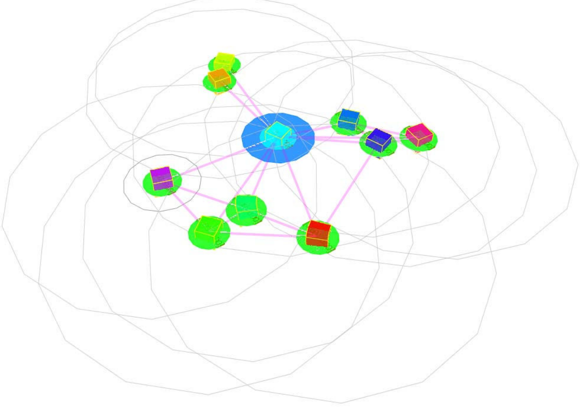

Fig. 2. 10 networked mobile robots in the Proto 3D simulator. The robot providing the network gateway is highlighted in the center (blue disc), lines between robots show radio connectivity, and circles around the robots depict communication range.

thus interested in the sparsest possible deployment, which still satisfies a certain minimal number of neighboring nodes for maintaining robustness.

We can summarize these requirements by the following constraint optimization problem

$$\max_P C(P) \tag{1}$$
$$s.t. \quad \alpha \le \|\mathcal{N}_i\| \le \beta, \qquad \forall i \in [1; n_0]$$
$$\overline{ig} < \infty, \qquad \exists g \in G$$

where $C(P)$ is the effective area being covered given the positions of the robots $P : \mathbb{R}^{2n_0}$ of all n_0 robots. $C(P)$ is given by the union of area covered by all robots

$$C(P) = \bigcup_{i \in [1;n_0]} A_i(p_i) \tag{2}$$

with A_i the coverage area of robot i at its position $p_i \in P$ and $\mathcal{N}_i$ is the set of robots in communcation range of node i, with $\|\mathcal{N}_i\|$ its cardinality. The set G comprises a set of gateway nodes to which a route must exist and $\overline{ig}$ denotes the minimal distance from robot i to robot g in hops.

A. Optimal Performance

The theoretical optimal coverage is bounded by $A_i n_0$. In this case, however, robots are not connected, as robot-to-robot distance must not exceed R (assuming a disc-shaped communication model). In case of a topology with at least one and not more than two links per robot, which allows maximal spread, each robot's coverage area A_i is reduced by $\frac{1}{3}\pi R^2 - \frac{\sqrt{3}}{4}R^2$ (see also the illustration in Figure 3, left), and the effective coverage area is given by

$$\bigcup_{i \in [1;n_0]} A_i \approx A_i n_0 - 2n_0 \left(\frac{1}{3}\pi R^2 - \frac{\sqrt{3}}{4} R^2 \right) \tag{3}$$

A near-optimal distribution which coverage comes close to (3) is illustrated in Figure 3, middle. When the topology requires at least 3 links per node, overlap between nodes increases and effective coverage is even smaller, which is illustrated in Figure 3, right.

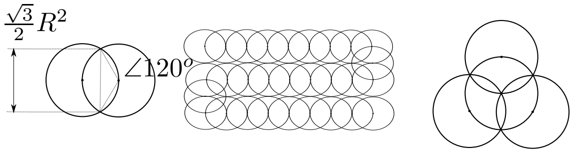

Fig. 3. Relation of effective area coverage and individual coverage for 2-link topologies (middle and left) and a 3-link topology (right).

IV. An Algorithm for Deployment and Optimization of a Wireless Ad-Hoc Network

In this section we develop the algortihm that provides a fully distributed solution to (1). Let $\mathcal{N}_i$ be the set of robots within communication range of robot i. We assume that each robot can estimate the number of other robots within signal range $\|\mathcal{N}_i\|$, avoid collisions with other robots and obstacles, and that the environment is bounded. We will first describe the algorithm (Section IV-A) and then propose a kinematic model (Section IV-C). Convergence of the algorithm is then proven using a probabilistic model (Section IV-D). Whereas the kinematic model bases on physical first principles and can be simulated in MATLAB, the probabilistic model is derived by associating probabilities with possible state transitions of the individual robot and allows us to capture the population dynamics of the whole swarm by a set of difference equations.

A. Deployment Algorithm

Maximizing $\cup A_i$ is equivalent to minimizing $\cap A_i$, i.e. the intersection between the coverage areas and can be obtained by having the robots randomly walk and mutually avoid each other for a sufficient amount of time. In a bounded environment, such behavior eventually leads to a uniform probability distribution for the robots in the environment. In order for satisfying the constraints of (1), we let robots only move if the constraints are *not* satisfied, i.e. the number of neighbors of a robot is not within the window given by $[\alpha; \beta]$, or a node is not connected to a gateway. By this, the random walk acts as an unbiased, distributed search on the configuration space. The algorithm is illustrated by an example in Figure 4 and summarized in pseudo-code (Algorithm 1). Robots move with speed v_i.

B. Proto implementation

The Proto implementation of this algorithm that will later be executed on the robots is as follows

```
(mov (if (> distance-to (gateway)) threshold)
  (brownian)
  (if (< (num-nbr) alpha)
    (brownian)
    (if (> (num-nbr) beta)
      (brownian)
      (tup 0 0)
    )
   )
```

Algorithm 1: DEPLOYMENT ALGORITHM (PSEUDO CODE)

```
Data: Number of robots $\|\mathcal{N}_i\|$ within range $R$ of robot $i$.
Result: A locally optimal deployment.
while $true$ do
    if not connected to gateway or $\|\mathcal{N}_i\| > \beta$ or $\|\mathcal{N}_i\| < \alpha$ then
        $heading = random([0; 2\pi[)$
        $speed = v_i$
    else
        speed=0
    move(speed,heading)
```

)

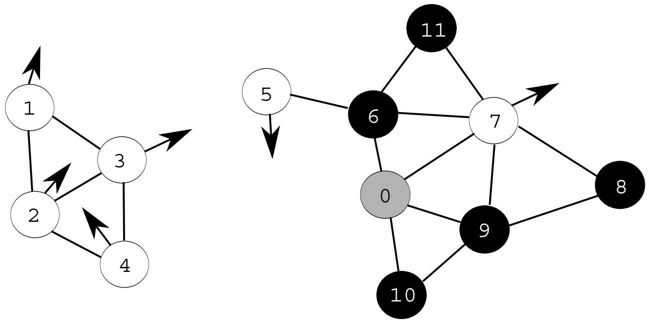

Fig. 4. Possible configuration during execution of the algorithm. White nodes are moving (an arrow indicating their random heading), whereas black and gray nodes are static. The gray node is the static gateway node, lines depict bi-directional one-hop connections. In this example, robots move if they have less than $\alpha = 2$ or more than $\beta = 4$ connections *or* are not connected to the gateway node by a finite number of hops. Nodes 1–4 are not connected to 0, whereas nodes 5 and 7 do not satisfy the connectedness constraints given by α and β, respectively.

Proto is a purely functional language written using s-expressions very similar to Scheme. A Proto expression produces a field that maps every point in space to a value. In our algorithm, the hop-count to the gateway (`distance-to`) and the number of neighbors (`num-nbr`) at the current location of the robot are mapped into a 2-tupel that defines direction and speed of the robot to move. If the number of neighbors is within the range given by `alpha` and `beta`, the expression evaluates to a Null-vector (`(tup 0 0)`). Otherwise—or when the distance to the gateway exceeds a certain threshold (usually the maximum number of robots)—the expression evaluates to a random vector (`brownian`) on the unit circle.

C. Kinematic Model

The kinematics of an individual node are given by

$$p_i(k+1) = p_i(k) + v_i \begin{pmatrix} \cos\phi_i \\ \sin\phi_i \end{pmatrix} \tag{4}$$

where $p_i(k)$ is the position of robot i at time interval k, v_i is a scalar speed, and ϕ_i is a random heading ($\phi_i \in [0; 2\pi[$). The algorithm is defined in discrete time and one time interval corresponds to the update speed T of the neighbor count estimate. In order not to penetrate the boundary of the environment $B \subset \mathbb{R}^2$ that can be detected using the on-board collision sensor, we set

$$(p_i(k+1) - p_i(k))\,|_B = 0 \tag{5}$$

and boundaries are avoided by choosing a new random heading ϕ_i.

The speed v_i of each robot is given by

$$v_i = \begin{cases} 0 & \text{if } \alpha \leq \|\mathcal{N}_i\| \leq \beta \wedge \overline{ig} < \infty, g \in G \\ v_i & \text{otherwise} \end{cases} \tag{6}$$

which reflects the constraints from (1).

D. Analysis of Convergence

We will now show convergence of Algorithm 1 by proving the convergence of a probabilistic model of the robotic system.

Theorem 1: A collective of robots with the behavior described in Algorithm 1 will converge into a single cluster, when $\beta > \alpha$, $\alpha, \beta \in \mathbb{Z}^+$ and $\beta < n_0 - 1$.

Proof: For convergence of the whole team into a single cluster, we need to show that the number of robots connecting to the cluster is larger than the number of robots leaving the cluster for all time intervals k. For this we will first develop a probabilistic model from the microscopic perspective of an individual agent (see for instance [11], [12]), which is inspired by the Master equation in physics [11], and models the probability for a robot to be in a certain state as a function of the other robots's states. We will then derive a set of macroscopic rate equations, which describe the average number of robots in a cluster. Convergence is finally shown by considering certain special cases, which help us highlight desired properties of the system.

1) Modeling Assumptions: Our analysis bases on the following assumptions. The environment is bounded, i.e. a randomly moving node will eventually visit every point in the environment with non-zero probability [13], and robots are randomly distributed in the environment according to a uniform distribution, i.e. the likelihood for encountering a specific robot is independent from its location in the environment. Although these assumptions seem to be restrictive, they correspond to the worst-case scenario of the proposed algorithm and predictions from models based on these assumptions can thus be understood as lower bound on the performance. Also, if we prove that robots converge from a random distribution, they will also converge from a centralized deployment. Using the assumption of spatial uniform distribution, we assume a constant probability p to encounter any other node in the environment during one time interval. The probability p is a function of the environment size, the sensor range, and the speed with which an individual node is moving through the environment and can either be measured in the real system or calibrated as proposed in [12].

2) Microscopic Model: In the algorithm described in Section IV-A, a node can be in one of the following states:

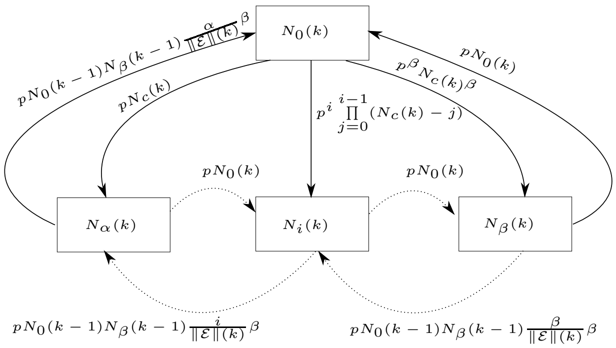
$N_0(k)$
$N_\alpha(k)$
$N_i(k)$
$N_\beta(k)$
$pN_0(k-1)N_\beta(k-1)\frac{\alpha}{\|\mathcal{E}\|(k)}\beta$
$pN_c(k)$
$p^i \prod_{j=0}^{i-1}(N_c(k)-j)$
$p^\beta N_c(k)\beta$
$pN_0(k)$
$pN_0(k)$
$pN_0(k)$
$pN_0(k-1)N_\beta(k-1)\frac{i}{\|\mathcal{E}\|(k)}\beta$
$pN_0(k-1)N_\beta(k-1)\frac{\beta}{\|\mathcal{E}\|(k)}\beta$

Fig. 5. Markov chain describing the degree distribution dynamics based on possible states of an individual node.

being mobile or being part of the cluster connected to the internet via any allowed number of neighbors (given by the parameters α and β for the lower and upper bound, respectively). Thus, the total number of states of an individual robot is bounded. The probabilistic finite state machine is summarized in Figure 5, and its state transition probabilities are developed as follows.

Given the probability p to encounter another robot (with $pn_0 < 1$), the probability to encounter one and only one out of N_c objects in the cluster is given by pN_c, and the probability for a mobile node to encounter exactly i other nodes in the same time-step is given by

$$p(i|0) = p^i \prod_{j=0}^{i-1}(N_c(k) - j), \qquad i \in [\alpha; \beta] \tag{7}$$

When a mobile node encounters i other nodes that are already part of the cluster, it will gain i neighbors. A mobile node joining the cluster, however, changes the status of all other robots it connects with. From the perspective of a robot with i connections, the probability that it is hit by a searching robot is $pN_0(k)$ with $N_0(k)$ the number of mobile robots at time interval k. Hence, the probability for a robot with i connections to get an additional connection during a time interval is given by

$$p(i+1|i) = pN_0(k), \qquad i \in [\alpha; \beta[\tag{8}$$

In the case of a robot already having β connections, this robot will leave the cluster with probability

$$p(0|\beta) = pN_0(k) \tag{9}$$

When this happens, all nodes previously connected to the leaving node will notice a lost connection in the next time interval. The probability for this event to happen to an individual node corresponds to the number of robots that left the cluster during the last time interval ($pN_0(k-1)$). As connections are removed from all nodes in the cluster with equal probability, the probability for an individual node to loose a connection needs to be normalized by the ratio of own connections and total connections in the cluster ($\frac{i}{\|\mathcal{E}\|(k)}$). Here, $\|\mathcal{E}\|(k)$ corresponds to the total number of edges in the system and is given by

$$\|\mathcal{E}\|(k) = \sum_{i \in [\alpha;\beta]} iN_i(k) \tag{10}$$

Hence, the probability for a robot with i connections to loose a connection is given by

$$p(i-1|i) = pN_0(k-1)N_\beta(k-1)\frac{i}{\|\mathcal{E}\|(k)}\beta, \qquad i \in]\alpha; \beta] \tag{11}$$

As robots only leave the cluster when they are exceeding β connections, each time a robot leaves, β other nodes loose a connection. This is reflected by the factors $N_\beta(k-1)$ and β in the above equation. For robots that have only α connections (the lower limit of allowed connections), removing a connection will let this robot resume search. Thus,

$$p(0|\alpha) = pN_0(k-1)N_\beta(k-1)\frac{l}{\|\mathcal{E}\|(k)}\beta \tag{12}$$

3) Macroscopic Model: Having established possible node states and state transition probabilities (see also Figure 5), we can derive the following set of rate equations:

The number of robots with i connections at time interval k is given by

$$\begin{aligned} N_i(k+1) = N_i(k) &+ p(i|0)N_0(k) \\ &+ p^i \prod_{j=0}^{i-1}(N_c(k) - j)N_{i-1}(k) \\ &+ pN_0(k-1)N_\beta(k-1)\frac{i+1}{\|\mathcal{E}\|(k)}\beta N_{i+1}(k) \\ &- pN_0(k)N_i(k) \\ &- pN_0(k-1)N_\beta(k-1)\frac{i}{\|\mathcal{E}\|(k)}\beta N_i(k) \end{aligned} \tag{13}$$

and is constructed by multiplying the state transition probabilities from and to state i (7), (8) and (11) with the number of robots in their original state. Equations for N_α, N_β, and N_0 can be constructed in a similar fashion.

4) Special cases: For $\alpha = \beta = 0$, nodes will not be able to connect to the cluster (as $\beta = 0$). For $\alpha = 0$ and $\beta \geq 1$, nodes will eventually join the cluster, and the rate of convergence will be a function of β. For low β, e.g. $\beta = 1$, the influx to $N_1(k)$ is given by $pN_1(k)N_0(k)$ and the outflux by $pN_0(k)N_1(k) + N_1(k)pN_0(k-1)\frac{1}{\|\mathcal{E}\|(k)}$. That is, nodes will leave the cluster at a higher rate than joining it. Similarly, we can see, that for all $\alpha = \beta = i$, $N_i = N_c$ and thus

$$p^i \prod_{j=0}^{i-1}(N_c(k) - j)N_0(k) < pN_0(k) + pN_0(k-1)\frac{i}{\|\mathcal{E}\|(k)}, \qquad i \in [0; n_0] \tag{14}$$

i.e. the inflow to the cluster is always lower than the outflow.

We will now consider the case $\alpha = 0$ and $\beta = n_0 - 1$. This corresponds to the scenario described in [9], who considers $\beta = \infty$, i.e. robots never resume motion once connected to the gateway. Our first observation is that $N_i(k)$ is rather

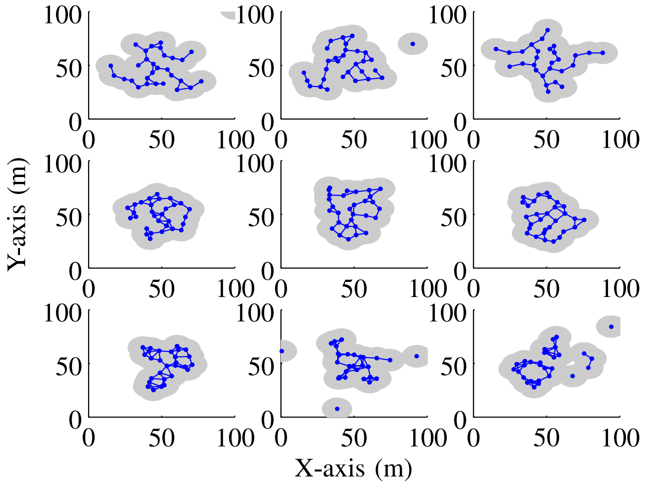

Fig. 6. Randomly selected deployments at time interval $k = 5000$ for a centralized initial configuration for $(\alpha = 1, \beta = 3)$ (top row), $(\alpha = 2, \beta = 4)$ (middle row), and $(\alpha = 3, \beta = 5)$ (bottom row), and robot speed $v_i = 1\frac{m}{T}$. Gray discs depict wirless range of an individual. Higher values for α and β lead to denser clusters, while topological optimization leads to instable behavior.

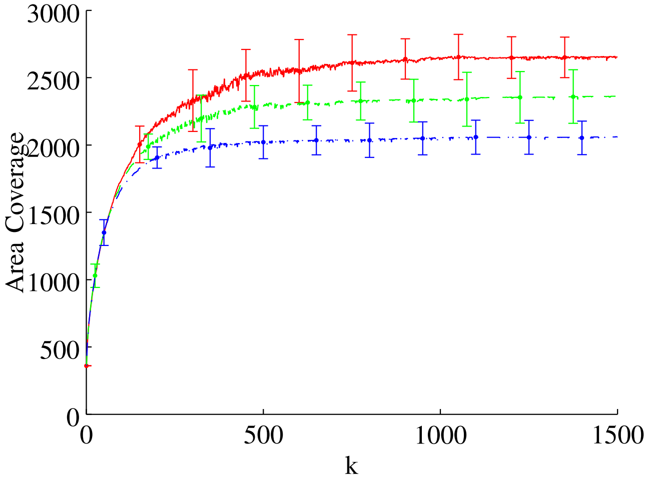

Fig. 7. Average coverage area vs. time for centralized deployment of 30 robots for $(\alpha = 2, \beta = 4)$, $(\alpha = 1, \beta = 5)$, $(\alpha = 1, \beta = 6)$, (top, middle, and bottom curve, respectively) for $v_i = 1\frac{m}{T}$. 100 runs, error bars show standard deviation. Broadening the window of allowable number of neighbors leads to faster convergence but lower coverage.

growing due to mobile robots connecting to less connected nodes than due to mobile nodes simultaneously connecting to i nodes as

$$pN_0(k) > p^i \prod_{j=0}^{i-1} (N_c(k) - j) \qquad (15)$$

with $p << 1/n_0$. This leads to

$$N_i(k) > N_{i+1}(k), \quad i \in [\alpha; \beta - 1] \qquad (16)$$

Consequently, if $\|\alpha - \beta\| > 0$, the outflow of the system will be always smaller than the inflow

$$pN_0(k)N_\beta(k) + pN_0(k-1)N_\beta(k-1)N_\alpha(k) < pN_0(k) \sum_{i\in[\alpha;\beta]} \frac{N_c!}{(N_c - i + 1)!} \qquad (17)$$

as $N_c(k) = \sum_{i\in[\alpha;\beta]} N_i(k)$ and $pN_0(k)N_\beta(k) > pN_0(k-1)N_\beta(k-1)$ for $N_\beta(0) = 0$. ■

V. RESULTS

We performed a series of simulations of the kinematic model given by (4)–(6) for both centralized and uniform initial deployments, as well as bounded and unbounded environments. In all our simulations, we use $n_0 = 30$ robots, an arena of $100 \cdot 100m^2$, and a wireless coverage area of a robot that is given by a circular disc with radius $R = 10m$.

Randomly selected topologies out of 100 simulations (kinematic model) at time $k = 5000$ for a centralized initial configuration are shown in Figure 6 and provide a qualitative idea of the resulting topologies. Stable clusters involving a large part of the population are usually reached within a few hundred time intervals.

For quantitatively studying the performance of the proposed algorithm, we measure the average area being covered by the wireless network, i.e. all robots that are connected to the gateway node by a finite number of hops, over 100 runs. Figure 7 shows results for increasingly large intervals of $[\alpha; \beta]$. We observe that increasing the range of allowed neighbors leads to slightly faster convergence at cost of decreasing coverage. This observation is well reflected in the probabilistic model as the probability to stop increases with the number of possible immobile states.

When exploring other intervals $[\alpha; \beta]$ we observe an increase in area coverage for lesser connected clusters, i.e. for lower values of α and β, as robots need to spread out more in order to satisfy the constraints.

We then simulated random initial distributions in order to test, whether the algorithm will indeed lead to convergence independently of the initial conditions. Qualitative results for random initial positions are depicted in Figure 8 for various settings of α and β. Figure 9 shows average coverage performance measured over 100 experiments as well as the theoretical optimal coverage $(\alpha = 1, \beta = 2)$. We observe that random initial deployments converge slower, but tend to lead to better overall coverage for sparse topologies as robots tend to connect to the cluster as soon as it becomes visible and thus better exploit the wireless signal range. This is not the case for dense topologies, which a) yield smaller cluster with a lower probability of being encountered and b) require the robots to come very close to the cluster before connecting (due to high values of α). We tested the algorithm also in *unbounded* environments for centralized and random deployments (within the 100m x 100m field as in the other experiments). Figure 10 shows box-plots of the performance over 100 experiments recorded at time interval $k = 5000$. Surprisingly, median coverage is only slightly lower in unbounded environments. These results suggest high robustness of the proposed algorithm also in open terrain.

We also conducted experiments in an indoor environment using real robots. Using a transmission power of 5 dBM,

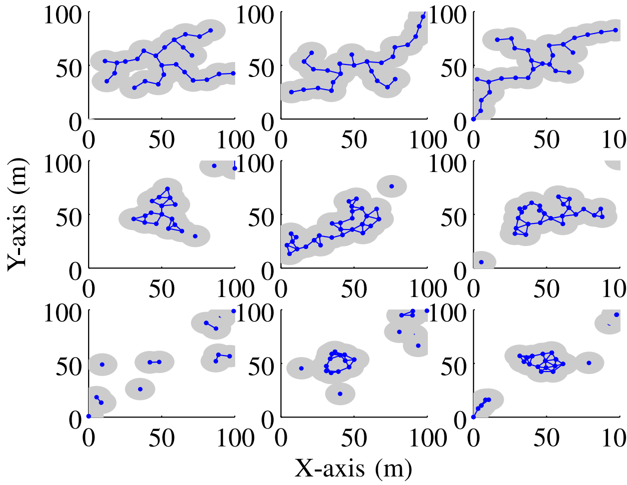

Fig. 8. Random samples showing the resulting topology at $k = 15000$ for random initial deployments for $(\alpha = 1, \beta = 3)$, $(\alpha = 2, \beta = 4)$, $(\alpha = 3, \beta = 5)$ (top, middle, and bottom rows, respectively).

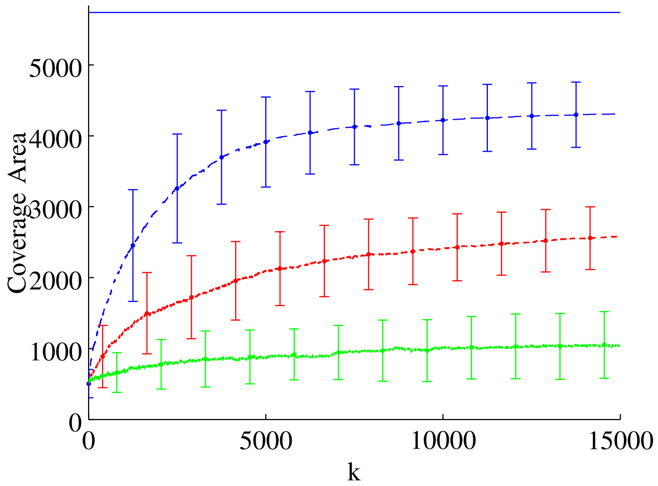

Fig. 9. Average coverage area vs. time for random deployment of 30 robots for $(\alpha = 1, \beta = 3)$, $(\alpha = 2, \beta = 4)$, $(\alpha = 3, \beta = 5)$ (top, middle, and bottom curve pairs, respectively) for *random* initial distributions and $v_i = 1\frac{m}{T}$. The more connections are required for robots to rest (high values of α), the slower the system converges. The straight line corresponds to coverage with a theoretical near-optimal topology.

the actual communication range varied between 10-20m. Figure 11 illustrates the performance of our algorithm and the Proto virtual machine. Robots were dispersed and shut down one after the other until we were left with two robots. As soon as the neighbor count reached the upper threshold, the two remaining robots instantaneously stopped moving and resumed motion only after the lower threshold was crossed. Figure 12 shows sample results from a centralized deployment of 9 robots on the ground floor of the MIT Stata center. In this experiment, convergence was achieved (all robots stopped) after approx. 35 minutes. The final coverage area was approximately $500m^2$. The qualitative behavior of the robots was very similar to that observed in simulation.

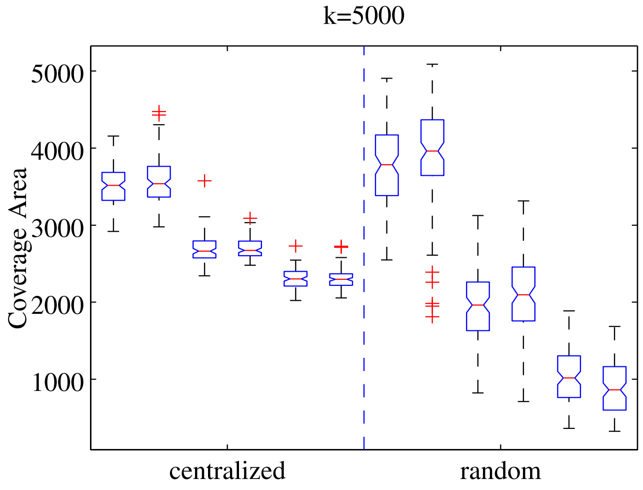

Fig. 10. The box-plots show the distribution of coverage at time interval $k = 5000$ for experiments in bounded and unbounded environments for $v_i = 1\frac{m}{T}$. Data for unbounded environments is shown on the left of each pair and data for bounded environments to the right. The three pairs to the left are from centralized deployments (Figures 7) and the three pairs to the right are from random deployments (Figure 9).

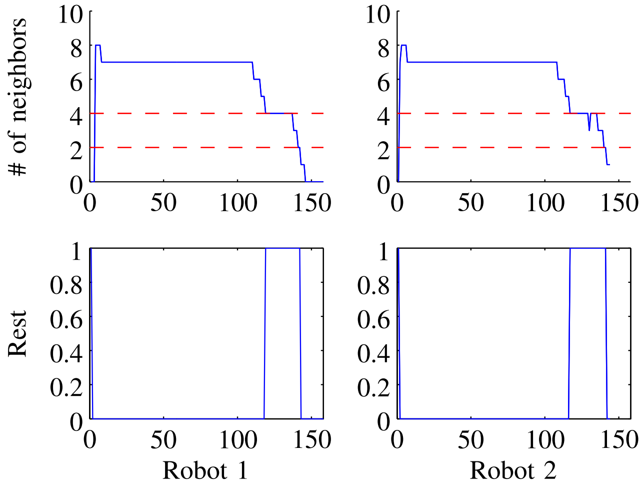

Fig. 11. Relation between number of robots within communication range and mobility for two robots. Robots have been systematically removed from the experiment, leading to rest of the robots when the number of remaining robots is within the allowable threshold.

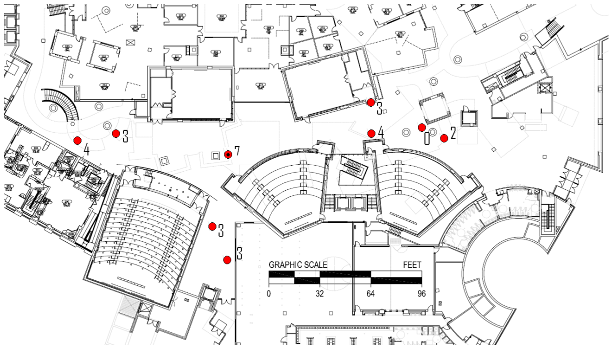

Fig. 12. Deployment of 9 robots on the ground floor of the MIT Stata Center after convergence. Numbers indicate the number of neighbors. The experiment lasted around 35 minutes and covers approximately $500m^2$. The robot in the center (marked with a black dot) is the static gateway and all robots were initially deployed around it. The second robot from the right showed a radio failure and is not counted by any other robot.

VI. Discussion

Extensive simulations show robust behavior with well shaped topologies, in particular for centralized deployments. Results from simulations with random deployments suggest good performance also for a real-world use case (e.g., where a human or robotic agent [14] deploys the robots according to some heuristic while exploring the environment).

In general we observe that changing the transmission power effects more the robustness towards external disturbances (e.g. people passing by) than the effective range. This makes the space required for conducting meaningful experiments rather large.

A. Probabilistic modeling

In addition to exhibiting guaranteed convergence, the dynamics of the probabilistic model are well in line with those of the kinematic simulations. For instance, for α being large, the model suggests slower convergence as the likelihood of a mobile node connecting to the cluster with α connections becomes low. Similarly, for large $\|\beta - \alpha\|$, not only the likelihood of sucessful connection to the cluster increases but the cluster will store more nodes, and thus increase speed of convergence.

A limitation of the proposed model is that measuring the degree distribution only indirectly captures the actual performance metric (area coverage). Also, the proposed state transition probabilities do not reflect the actual embodiement of the cluster. For instance, robots at the boundary of the cluster will have a much higher probability of connecting to a bypassing mobile robot. As all robots are connected, however, state transitions of the boundary robots will quickly propagate to the inside of the cluster. In order to capture this effect, the dynamics of the system might be better modeled by a spatial model, which describes the number of robots i hops away from the center or captures the spatial probability density function of the whole system.

VII. Conclusion

We studied a wireless coverage algorithm which poses minimalist requirements on the robotic hardware, namely knowledge of the number of wireless links and bumper sensors for collision avoidance. Parameters of the algorithm are lower and upper bounds on the desired connectivity of each individual node and the robot speed during deployment.

We explore properties of the algorithm at four different model abstraction levels. Kinematic simulation with a disc-shape communication model and a probabilistic, macroscopic model, which allows us to formally prove convergence. Convergence of the algorithm shows to be extremely robust and independent of the initial deployment of robots, and whether the environment is bounded or unbounded. Although the proposed algorithm never achieves the theoretical optimal coverage, all of 100 simulations with topologies with 1 to 3 links per node are within 52% and 81% of optimal performance, and half of the simulations at 64% of the optimum. Consequently, we observe similar performance also in a real-world deployment (given sufficient locomotion capabilities to explore a large enough environment).

In the future, we are interested to study how additional ressources such as odometry, global localization or increased communication will affect performance. Notably, we will explore odometry and global localization for better-than-random search and communication for locally estimating the global topology, which might allow for actively supporting weak connections in the graph or increase the speed of convergence by more directed motion. We are also interested in considering problems in which the gateways themselves are mobile or where constraints are extended to provide tethering

Acknowledgments

The authors would like to thank A. Derbakova, J. Eriksson, and A. Swaminathan for help with implementing the Proto virtual machine on the Atheros MIPS/Create platform and J. Beal for helpful discussion on the Proto language.

References

[1] H. Tanner and C. Abdallah, Eds., *IEEE Control Systems Magazine, Special Issue on 'Complex Networks and Interconnected Control Systems'*, August 2007.

[2] E. Stump, A. Jadbabaie, and V. Kumar, "Connectivity managment in mobile robot teams," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Pasadena, CA, 2008, pp. 1525–1530.

[3] A. Howard, M. Mataric, and G. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots Special Issue on Intelligent Embedded Systems*, vol. 13, no. 2, pp. 113–126, 2002.

[4] N. Heo and P. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," in *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 3, March 2003, pp. 1597–1602.

[5] S. Poduri and G. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE International Conference on Robotics and Automation*, New Orleans, LA, May 2004, pp. 165–172.

[6] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[7] K. Støy, "Using situated communication in distributed autonomous mobile robots," in *Proc. of the 7th Scandinavian Conference on Artificial Intelligence (SCAI-7)*, Odense, Denmark, 2001, pp. 44–52.

[8] J. Nembrini, A. Winfield, and C. Melhuish, "Minimalist coherent swarming of wireless connected autonomous mobile robots," in *Proc. Simulation of Artificial Behaviour SAB 2002*, Edinburgh, UK, September 2002, pp. 273–382.

[9] S. Poduri and G. Sukhatme, "Latency analysis of coalescence in robot groups," in *IEEE International Conference on Robotics and Automation*, Rome, Italy, May 2007, pp. 3295–3300.

[10] J. Beal and J. Bachrach, "Infrastructure for engineered emergence on sensor/actuator networks," *IEEE Intelligent Systems*, vol. 21, no. 2, pp. 10–19, 2006.

[11] K. Lerman, A. Martinoli, and A. Galystan, "A review of probabilistic macroscopic models for swarm robotic systems," in *Proc. of the SAB 2004 Workshop on Swarm Robotics*, Santa Monica, CA, USA, 2005, pp. 143–152, Lecture Notes in Computer Science Vol. 3342, Springer-Verlag, Berlin.

[12] A. Martinoli, K. Easton, and W. Agassounon, "Modeling of swarm robotic systems: A case study in collaborative distributed manipulation," *Int. J. of Robotics Research*, vol. 23, no. 4, pp. 415–436, 2004.

[13] H. C. Berg, *Random Walks in Biology*. Princeton University Press, 1983.

[14] M. Batalin and G. Sukhatme, "The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 661–675, 2007.