Route in Mobile WSN and Get Self-deployment for Free

Kévin Huguenin[1], Anne-Marie Kermarrec[2], and Eric Fleury[3]

[1] IRISA / ENS Cachan
[2] INRIA Rennes - Bretagne Atlantique
[3] INRIA Lyon - Rhônes-Alpes / ENS Lyon

Abstract. We consider a system consisting of a set of mobile sensors. They are disseminated in a region of interest and their mobility is controlled (as opposed to mobility imposed by the entity on which they are embedded). A routing protocol in this context enables any point of the region to be reached starting from any node, regardless of the initial sensor deployment. This operation involves message forwarding and/or sensor motion. In this paper we present GRASP, a GReedy stAteless Routing Protocol for mobile wireless sensor networks (WSN). GRASP is simple and independent from the underlying communication model, but still provides results close to the optimal, with respect to the self-deployment of sensors over a given region. It ensures that *(i)* routing is always possible in a mobile WSN irrespective of the number of sensors, and *(ii)* above a given number of sensors in a considered zone the protocol eventually enables the routing to no longer require sensors to move, which yields to self-deployment. With GRASP, sensors autonomously reach a stable full coverage following geometrical patterns. This requires only 1.5 times the optimal number of sensors to cover a region. A theoretical analysis of convergence proves these properties. Simulation results matching the analysis are also presented.

1 Introduction and Background

Consider a troop of human agents deployed in a region to accomplish a mission and assisted by a set of networked mobile devices, whose mobility is controlled (as opposed to mobility imposed by the entity on which they are embedded). An agent may order a mobile device close to him to perform an action at a given location of the region. Note that the action can eventually be performed by *any* node of the system. This is achieved by routing to this specific point. To perform routing a node can either *(i)* perform the action itself if possible, *(ii)* forward the request to another node or *(iii)* move. Such systems have numerous applications, a typical one being a situation where a military unit secures a sensible zone with the help of mobile sensors able to move, detects enemies (sensors) and raises the alarm. Another possible application is the case of a brigade of firefighters with mobile air-pressurized water autonomous fire extinguisher. It enables robots to be deployed in a zone where human cannot be yet. Once the robots have secured

B. Krishnamachari et al. (Eds.): DCOSS 2009, LNCS 5516, pp. 201–215, 2009.

this zone (say extinguish the fire), the robots can then self-deploy to monitor the zone.

Similarly to traditional static sensor networks, the *coverage* of the region and the *connectivity* of the system reflect the quality of the deployment. Even though any point can be reached irrespective of the deployment by having nodes move, the movements incur delays. A routing algorithm must be evaluated against two metrics: *reactivity* in a *dynamic* scenario (when the troop progresses in the region, referred as the *transient state*) and *self-deployment* in a stable scenario (when the troop secures the zone inside which it is deployed, referred as the *steady state*).

The aforementioned applications require a routing algorithm capable of leveraging the node's communication and sensing capacities as well as the nodes's mobility to fulfill a request from a user, at any position. On one hand, a large amount of research has been devoted to finding deployments that ensures the connected coverage of a region of interest (ROI), thus leading to the design of optimal configurations [1]. Although self-deployment techniques allow us to autonomously reach such configurations by the mean of robotic sensors [2], they require an a-priori knowledge of the ROI at each node and are therefore not able to deal with evolving region and dynamic sensor relocation. Moreover, a traditional static (i.e. which does not leverage the node mobility) routing algorithm may not be able to cope with disconnected network or sensing node outside of the network scope. In [3], Butler and Rus considered a similar application. Assuming that sensors are notified of events occurring in their environment, Butler and Rus proposed an efficient decentralized self-deployment algorithm making the nodes converge reactively in the most interesting portions of the region while still ensuring full coverage. Yet powerful, this algorithms may not match our applicative context since *(i)* it requires an underlying protocol to advertise to the nodes the location of relevant events and *(ii)* it does not address the problem of routing in this context.

On the other hand, a lot of research on routing in disconnected networks and Delay Tolerant Networking [4] (DTN) in Mobile Ad-Hoc Network [5] (MANET) have been conducted. That led to the design of efficient routing algorithms and powerful mobile computing systems, such as Car-Tel [6]. Yet, except from message ferrying approaches – which exploit mobile nodes with uncontrolled but predictable mobility [7,8], and MORA[9] - a motion planning based routing algorithm – which exploits the *controlled* mobility of a small set of autonomous agents; these algorithms cope with the node *uncontrolled* mobility rather than leveraging it for routing purposes.

To the best of our knowledge, the two research topics of mobile routing and self-deployment have been mostly studied independently. Our claim is that using a routing algorithm leveraging the node mobility allows to cope with dynamic ROI, inefficient deployment of the nodes and under-dimensioning (i.e. the number of sensors is not sufficient to coverage the ROI in a connected way). In addition, it provides self-deployment for free in *steady state*.

Our contribution is two-fold. First we present Grasp, a simple routing algorithm acting only with local knowledge of the network and no knowledge of the ROI (i.e., the nodes do not know its size, its shape or its borders) and prove that beyond its simplicity Grasp ensures both sensing request fullfilment with probability one regardless of the network configuration, and a convergence toward connected-coverage of the ROI with a required number of node close to the optimal in steady state. Then we investigate the properties required for a generic mobile routing algorithm to provide self-deployment. We demonstrate by considering practical matters that Grasp may be used in a real-environment. We focus on 2D ROIs. The analysis of 1D shows that Grasp is optimal, details are provided in [10].

The rest of the paper is organized as follow: Section 2 presents the design rationale behind Grasp along with the detailed algorithm. Section 3 provides a theoretical analysis of Grasp with respect to the self-deployment properties. Section 4 presents experimental results obtained through computer simulations that match the theoretical analysis. This section also gives a performance analysis of Grasp with respect to *(i)* its efficiency in terms of routing delays and energy, and *(ii)* its impact on the network topology in terms of self-deployment. Sections 5 tackles the practicality of Grasp by suggesting an algorithm to handle concurrent message routing and a sleep-wakeup scheme leveraging the network geometry resulting from Grasp to increase the system lifetime. Finally, we provide in Section 6 a list of perspectives and on-going work to increase Grasp performance.

2 GRASP: A Routing Algorithm for Mobile WSNs

As stated in the introduction, neither traditional routing algorithms nor self-deployment techniques can be used for the targeted applications. In this section we present Grasp, a routing algorithm leveraging the node mobility to cope with dynamic ROIs and sparse or non-homogeneous deployments.

2.1 System Model

We consider a network of mobile entities with wireless communication capabilities deployed in an obstacle-free region. We assume a disc model for sensing (i.e., a node is able to sample its environment up to a distance r_s from its current position) and symmetric reliable communication links. In the sequel we denote by routing the action of making any node fulfill a sampling request, i.e., sense at a given location in the ROI, emitted at a given node in the network. We further assume that nodes are able to orientate and localize themselves inside the ROI by the use of a compass and a localization system such as a GPS or a distributed location algorithm [11]. In addition, we assume that they also know their neighbors' coordinates, using for instance periodic beacons.

2.2 Design Rationale

The design rationale behind GRASP can be explained through the analogy with a soccer game. In a game, a set of mobile intelligent entities, namely the *players*, are deployed on a rectangle area, the *pitch*, and collaborate in order to deliver a packet, the *ball*, at a given position, the *goal*. To succeed, the players can either run or pass the ball to a team-mate provided that the distance between them is not too large. Obviously, passing the ball to a team-mate is less tiring than running to put the ball in the goal. In this context, the energy constraint is that the players must keep the ability to move until the end of the game. On one hand, the players must pass the ball as often as possible so as to save their energy, but on the other hand, at some point, a player's reachable team-mates may all be in worst position than himself to reach the goal. Two questions arise: assuming a limited view of the game and limited passing capabilities, *(i)* "to which of its reachable team-mates should a player pass the ball?", and *(ii)* "when should a player move instead of passing?"

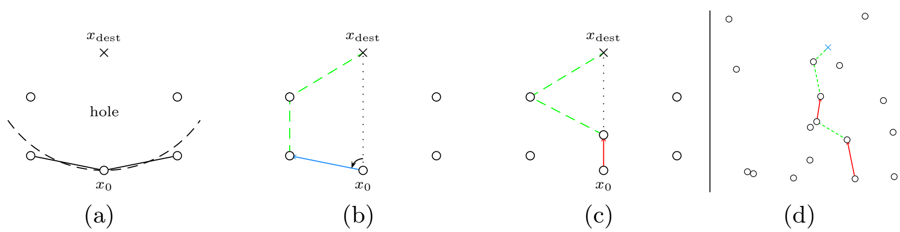


Fig. 1. Illustration of a *routing hole* in a WSN. (a) Node x_0 is closer to the destination x_{dest} than its neighbors making greedy routing fail. (b) Face-based routing algorithms route along the edges bordering the void while (c) Grasp leverages the node mobility to move toward the destination until greedy forwarding is possible. (d) Sample path using GRASP. Dashed lines denote *greedy forwarding* and solid lines and arrows denote respectively hole circumventing and moving.

Following this analogy, we propose a simple geographic routing algorithm leveraging nodes mobility to transparently cross holes in the topology. Nodes act greedily for both forwarding and moving: the distance between the current position and the destination should always be decreasing. If the destination lies in the sensing disc of the node in charge of the request (i.e., its distance to the destination is lower than r_s), the node fulfills the request itself. If not, it can either forward the message or move as follow:

Forward: using local information on its neighbors' position, which is updated by means of periodic beacons, the node in charge of the message picks the closest node, *and closer than itself*, to the destination – if any – and forwards.

Move: otherwise, the node starts moving in straight line toward the destination until it can either forward the message to a node closer than itself to the destination or sense itself at the destination. The node stops and performs the adequate action.

Algorithm 1 gives a detailed pseudo-code version of GRASP. Figures 1(a)-1(c) illustrate the way GRASP deals with routing holes as compared to traditional static geographical routing algorithms [12] (such as GPSR [13]) and Figure 1(d) shows a sample path – combining both forwarding and moving – used by GRASP to deliver a message in a sparse 2D mobile WSN.

Algorithm 1. GRASP: A routing algorithm for mobile WSN

Input: upon reception of a sampling request (for position x_{dest}) at node x_0
  while $d(x_0, x_{\text{dest}}) > r_s$ and $\nexists x \in \text{neighborhood}(x_0) \mid d(x, x_{\text{dest}}) < d(x, x_0)$
    move toward x_{dest}
  end while
  if $d(x_0, x_{\text{dest}}) \le r_s$ **then**
    sample x_{dest}
  else {exists a node in x_0's neighborhood closer to x_{dest} than x_0}
    forward to $\underset{x \in \text{neighborhood}(x_0)}{\operatorname{argmin}} d(x, x_{\text{dest}})$
  end if

The strengths of the algorithm described above are its simplicity and the resulting characteristics. First, no assumption is made on the radio communication model. More specifically, the traditional unit disc model is *not* assumed and the communication range is *not* explicitly used by the algorithm. Actions are only based on a pure localized and distributed information built from periodic beacons advertising the identifier and the position of a node.

The second important characteristic relates to the deployment: GRASP acts with zero knowledge on the ROI. GRASP is purely distributed and decentralized. Its ultimate goal is to allow an action to be performed at any location of the ROI. The deployment of the nodes is not explicitly controlled by the geographical shape of the ROI but by the application requirements. If an agent needs an intervention at specific positions, it sends requests toward these positions. Nodes route requests and thus potentially move in order to satisfy them. The deployment of the nodes is dynamic and self adapts to the shape and to any potential evolution of the ROI.

Assuming no packet loss and no node failure GRASP provides sampling request fullfilment with probability one, regardless of the number of sensors: at each step the distance between the node in charge of the message and the destination is reduced either by moving toward the destination or by forwarding the message to a closer node. Therefore, GRASP outperforms traditional geographic routing as it takes benefit of the nodes mobility to overcome dead-end routes. On the other hand, GRASP transparently increases the coverage by automatically filling routing holes. Intuitively, a node is required to move when the area between itself and the destination does not contain any other node. Therefore, moving toward the destination fills the routing hole. More concretely, GRASP offers a spreading property in the sense that, in addition to filling routing holes, when making a node move, it may not get closer to any other node than a given distance called *repulsion radius*, close to the communication radius (a closed form expression of the radius is given in Section 3.1). Moreover, the deployment of nodes is on demand. Such a reactive behavior does not require any specific hole detection or a pre-deployment computation phase. In addition, all properties of GRASP hold with evolving, in shape and size, regions of interest without explicitly requiring to be aware of such changes.

Note that we do not establish one rigid path from one node to another: routing a packet from the same source node to the same destination may require some nodes to move. This is not a burden as most of the targeted applications require to send an order without expecting an answer right away. For instance, the ultimate goal of our example application is to have *any* node check at a given location and raise an alarm if required but not to send back any specific piece of information.

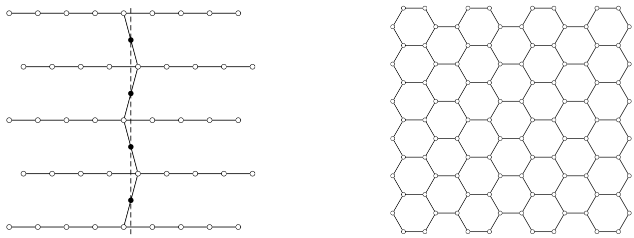
3 Theoretical Analysis

In this section, we present a theoretical analysis of our routing protocol for mobile WSN with respect to self-deployment.

We denote by an *stable deployment* a network configuration in which any point of the ROI can be reached (by forwarding and then sensing), without requiring any node to move, while using the greedy routing algorithm presented in Section 2. Due to the greedy nature of the forwarding algorithm, such a configuration should provide *full greedy-connectivity.* Our analysis considers the case where the sensing radius r_s is equal to the communication range r_c. We make assumption on the disk model only to derive analytical results. Note that relaxing the assumption $r_s = r_c = R$ impacts on the optimal configuration but the general sketch of the proofs still holds. The purpose of this paper is not to study exhaustively all the possible ratio values between r_s and r_c but to present a formal framework for GRASP. We prove that GRASP converges, to a sub-optimal *stable configuration* (i.e. a stable configuration using the 1.5 times the minimum number of nodes) provided that the number of nodes is sufficient.

3.1 Background

A recent study by of Iyengar *et al.* [14] explores the problem of the optimal deployment of a static WSN ensuring a *connected-coverage* of a region (*i.e.*, full coverage with full connectivity) focusing on the case $r_s = r_c = R$. Using geometric considerations, they derive a lower-bound on the optimal node density to cover a zone in a connected way and they propose a strip-based configuration which tightly approaches the bound. In addition the strip-based configuration is asymptotically optimal. This configuration is composed of horizontal connected strips spaced by $\left(1 + \sqrt{3}/2\right) R$. This way, any two nodes on the same line can communicate. On the other hand, a vertical strip of nodes connects the horizontal



(a) Strip deployment. (b) Hexagonal deployment.

Fig. 2. Node deployments ensuring (a) asymptotically optimal connected-coverage and (b) optimal greedy connected-coverage

strips together ensuring the full connectivity of the network. However, a greedy geographic algorithm may not be able to reach any point of the ROI. Figure 2(a) depicts the strip-based deployment. In [1], Bai *et al.* extended those results by proving the asymptotic optimality of the strip-based deployment pattern for any $r_c/r_s < \sqrt{3}$ (not only $r_c = r_s$).

In their work, Iyengar *et al.* considered full connectivity which characterizes a configuration where there *exists* a path between any two nodes. However, our work focuses on network deployment where a greedy geographic algorithm *can find* a path between any two nodes. The full greedy connectivity can be formalized as follow: for any destination point of the ROI, any node x_0 in the network can communicate with a node x_1 closer to the destination than x_0. Based on this definition, it can be proved [10] that the hexagonal lattice (see Figure 2(b)) is the optimal deployment ensuring both full greedy-connectivity and the required density of sensors to cover the area using this mesh is $4\sqrt{3}/9R^2$. In addition this configuration ensures full coverage of the ROI provided that the sensing radius is greater or equal to the communication radius.

3.2 Proof of Convergence

A first evidence is that nodes can not get closer from each other than a given characteristic distance δ called the *repulsion radius*. Consider a node x_0 moving toward a point of the ROI x_{dest}. Node x_0 crosses the communication disc of a node x_1 if its distance to the destination when it enters in x_1's communication disc is smaller than the distance between x_1 and x_{dest}. That is, using the notations in Figure 3(a):

$$y \geq \sqrt{R^2 - \delta^2} + \sqrt{y^2 - \delta^2} \ .$$

This inequality holds for any $y \leq R \ / \ \left(2\sqrt{1-(\delta/R)^2}\right)$. The condition $y > R$ implies $\delta > \frac{\sqrt{3}}{2} \cdot R$. Therefore, even if a node x_0 may cross the R-disc centered on a node x_1 when traveling toward its destination x_{dest}, the distance between x_0 and x_1 stays always larger than $\frac{\sqrt{3}}{2} \cdot R$. We denote by *repulsion radius* this minimum distance. Note that the repulsion radius is smaller than the communication radius. One may thus model all nodes by physical soft balls of radius ranging from $\frac{\sqrt{3}}{2}\frac{R}{2}$ to $\frac{R}{2}$.

Based on the physical model analogy, running GRASP on a mobile WSN can be thought of as packing a set of balls inside a given frame [15]. As one may recall, the intrinsic action of GRASP on mobile nodes is rather to push them than to pack them. Each couple (x_1, x_2) of two adjacent balls $(d(x_1, x_2) \leq R)$ presents two attraction sites (see Figure 3(b).) An attraction site is a place where the probability to move, in order to sample a location behind the line x_1x_2, is null. Note that such a position also minimizes the size of the associated Voronoï region (the set of destinations making a node located at this position move to fulfill a sampling request.) If a ball x_0 is not located in an attraction site, in other words, if it is not adjacent to both x_1 and x_2 but only to one of them (x_2 without loss of generality) then for each sampling location behind the line x_1x_2, it has a strictly positive probability to move. More precisely, if the sampling location is located on the left of the median of $[x_0x_2]$ and behind the line x_1x_2, the probability to move is non null (related to the hatched area on Figure 3(c).) Moreover, moving toward a sampling location inside this area pushes the node closer to the attraction site and the ball x_0 then touches x_1. During the process, additional nodes may also arrive and stick to one another. Eventually, all nodes will converge to an attraction site with no more possibility to escape. Such final configuration is the well-known stable triangular lattice. Note that we do not claim that only triangular tiles can be formed but rather that they are the most likely to be formed. More specifically, hexagonal tiles allow greedy routing (as shown in the previous paragraph) and require less nodes but they are not stable. Effectively, small deviations on the nodes' position in an hexagonal tile may lead to a reconfiguration of the tile into a triangular-like tile: a node routing toward the opposite node in the hexagon would move and stop near the center of the hexagon forming equilateral triangles.

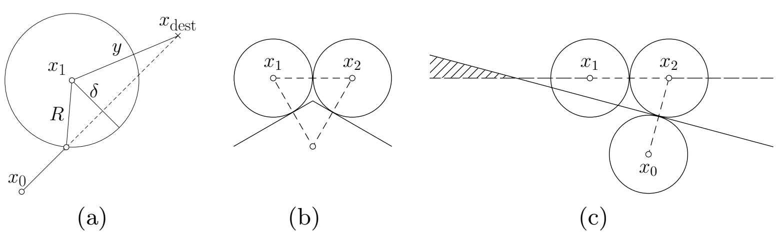


Fig. 3. Repulsion radius and attraction sites: (a) two nodes running GRASP cannot get closer than $\delta = \frac{\sqrt{3}}{2}R$; (b) the two points which forms, together with x_1 and x_2, an equilateral triangle are attraction sites where a node does not need to move to route a message behind the line (x_1x_2). (c) Routing toward a destination in the hatched area forces x_0 to move. Plain lines delimit the Voronoï region associated to the nodes.

4 Simulation Results

In this section we present the results of computer simulations and show that they closely match the theoretical analysis conducted.

4.1 Experimental Setup

In order to stretch the self-deployment capabilities of GRASP, sensor nodes are deployed uniformly at random in a restricted area of the ROI. The considered region is a square folded into a torus shape (note that we use a toric ROI for the sake of simplicity, using a square ROI only causes minor changes on the geometric deployment pattern near the borders.) It requires 500 nodes to be connected-covered, with respect to the expressions of optimal density given in Section 3. In order to assess the threshold beyond which GRASP converges, we consider several values of $\rho = N/N_{\text{opt}}$, close to 1.5. This enables us to study the behavior of GRASP under slight under or over-dimensioning in the number of nodes. The results can be used to evaluate the impact of failures on the deployment. The system dynamic is controlled, *via* GRASP, by message emission. We use a constant message emission rate λ, uniform amongst the nodes. Thus, during one time unit $\lambda \cdot N$ messages are routed from a randomly chosen node drawn from a uniform distribution on the full set of nodes to a destination drawn from a uniform distribution on the entire ROI. For the sake of simplicity, we consider the routing of *one message at a time*, the case of concurrent routing being tackled in Section 5.1. In this context, we evaluate GRASP's behavior and convergence properties along the following metrics:

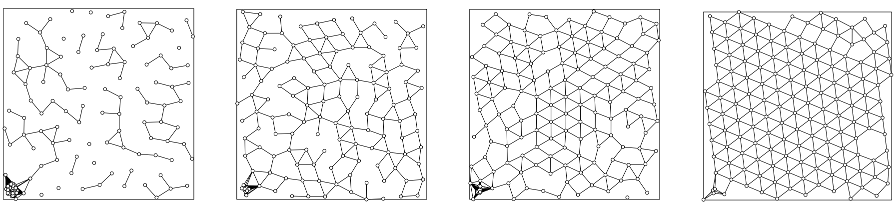
Average distance covered per node to deliver a message $\tilde{d}$: this metric reflects the uniformity of the node deployment and thus the distance between the current configuration and the optimal deployment. An optimal deployment being a network configuration where any point of the ROI can be reached without moving, $\tilde{d}$ should decrease to zero as time tends to infinity (assuming a constant emission rate λ) provided that the number of nodes is sufficient. In addition, assuming that a move is much more time-consuming than a wireless transmission, this metric reflects *(i)* the global mechanical (i.e., the movement) energy consumption of the system and *(ii)* the average delivery delay.

Distribution of the average distance covered by a node to deliver a message p_D: this metric brings additional information on the network load. More specifically, it reflects how the mechanical energy consumption is distributed amongst the nodes. The probability density function of d in a stable configuration is an impulse, located at zero.

Probability that at least one node moves to deliver a message p_m: the order of magnitude of the delivery delay is dependent on whether the messages are delivered using only wireless communication or by moving to cross routing holes, regardless of the distance covered. Thus, p_m is a good indicator of the system quality of service provided by the system, dual to $\tilde{d}$.

4.2 Evaluation

We present simulation results obtained by averaging over 25 runs of *Monte-Carlo* simulations. In each configuration, the metrics are computed by running $20 \times N$ independent simulations. For instance, $\tilde{d}$ is evaluated by averaging the distance covered by a nodes to deliver $20 \times N$ random messages.



(a) $t = 1$ (b) $t = 5$ (c) $t = 50$ (d) $t = 1000$

Fig. 4. Evolution of the network topology over time. A hundred nodes are initially uniformly deployed on a small area (1/100th) of a 2D toric ROI. The message emission rate is set to $\lambda = 10$. The plain lines represent the communication graph. For the sake of simplicity, edges between border vertices are not represented. Note the existence of an hexagonal tile and a few square tiles in the communication graph.

Evolution of the network topology. Figure 4 depicts the evolution over time of the network topology. To match our theoretical analysis, we consider in this experiment 1.5 times the number of nodes required for an optimal deployment with the minimum number of nodes. We observe that starting from a localized initial deployment over the ROI, GRASP converges to a stable deployment (i.e., connected-coverage). Figure 4(a) shows that the first stages of the deployment spread the nodes inside the entire ROI. Figures 4(b)-4(d) illustrate the aggregation process and the final configuration described in Section 3.1.

Average distance covered. Figure 5(a) depicts the evolution of the average distance covered by node to deliver a message $\tilde{d}$. We consider a network with a number of nodes equal to $\rho = 1$, 1.25, 1.5 and 1.75 times the optimal. As expected from Section 3.1, $\tilde{d}$ tends to a non-null limit for $\rho < 1.5$ and to zero otherwise. This matches the theoretical results, since the triangular lattice formed by GRASP requires 1.5 the minimum number of nodes to ensure greedy connected-coverage. Yet after 20 iterations, $\tilde{d} \approx 4.10^{-4}R$ (when $\rho = 1.5$ and $2.10^{-4}R$ when $\rho = 1.75$) which is negligible: for a radius of 100m, each node would be expected to move on average of 4cm. Note that using only 17% (1.75/1.5) more nodes than needed reduces the average distance covered by a node by 50%.

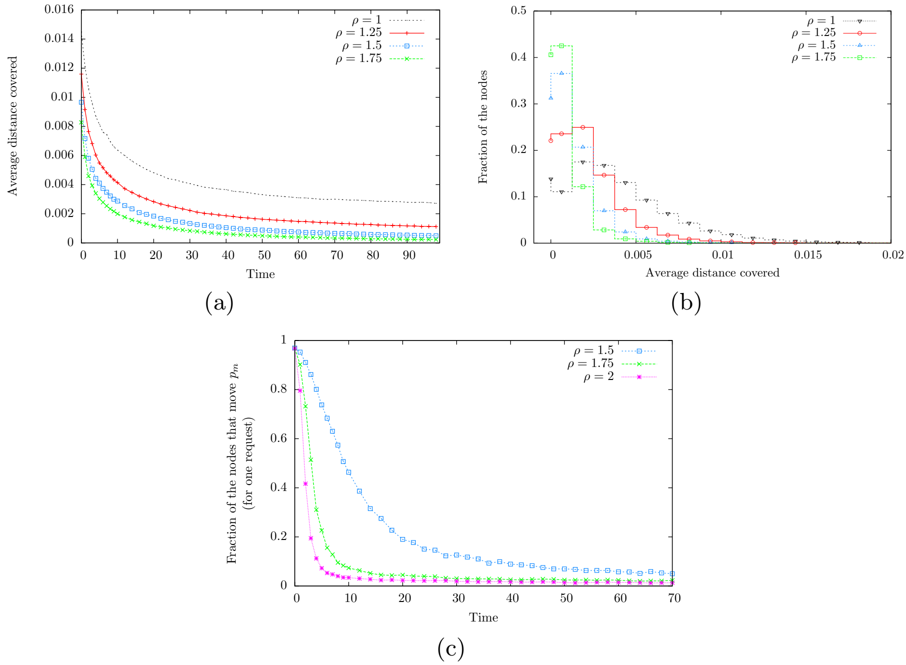


Fig. 5. Evaluation of Grasp in a 2D toric ROI: (a) average distance covered per node (relatively to R) as a function of time ($\lambda = 1$) ; (b) load distribution in the systems with respect to the average distance to be covered by a node to deliver a message (relatively to R) ; and (c) the probability that Grasp uses mobility to deliver a message as a function of time ($\lambda = 10$)

Network load. Figure 5(b) depicts the distribution of distances that nodes need to cover to deliver a packet after 20 time units. When the number of nodes is sufficient for Grasp to converge (i.e., $\rho \geq 1.5$), the curves are decreasing and show that convergence is in progress. The high fraction of nodes which never move (up to 30% for $\rho = 1.5$) together with the shape of the histogram reflects the existence of large greedy-connected components with low redundancy inside each of them: the nodes inside the greedy connected components never move and the ones on their border moves to ensure connectivity between the components. The fact that the distance covered by the node is very low (at max 1% of the communication radius for $\rho = 1.5$) implies that the components are spatially extended, reflecting a low redundancy (with respect to the initial deployment where most of the nodes never move but where the ones that move cover on average 1/4th of the ROI.) When the number of nodes is not sufficient for Grasp to converge (i.e. $\rho < 1.5$), the network load has a bell-shape with a maximum at $d = 25.10^{-5}R$ (for $\rho = 1$), a heavy tail for the large values and a non-negligible fraction of *still* nodes (15% for $\rho = 1$.) Therefore the network load in terms of distance covered is well balanced between the nodes. Thus the energy needed for routing is evenly shared between a large fraction of the nodes resulting in an extended lifetime of the system.

Probability to move. Figure 5(b) presents the probability that at least one node moves to deliver a message for a number of nodes sufficient for GRASP to converge. With the minimum number of nodes ($\rho = 1.5$), p_m decreases quickly to 20% and then converges slowly to zero. For slightly higher values of ρ, the first decreasing stage is drastically speeded-up: $p_m \approx 2\%$ after less than 20 iterations.

Summary

Simulations confirmed the theoretical results presented in Section 3. GRASP requires only 1.5 the minimum requested number of nodes to converge resulting in a triangular lattice deployment. Not only do we believe that this is a reasonable bound but this increased number of nodes is leveraged in several ways. First, the triangular lattice provides, over the other regular lattices (including the hexagonal optimal deployment), an increased resilience to failure as each node is provided with two potential neighbors in any direction. Second, as we will explain in Section 5.2, the fact that the triangular lattice is a subgraph of the hexagonal one can be exploited by a clever sleep-wakeup scheme to increase the lifetime of the system of 50%, fully justifying the 1.5 number of nodes over the minimum.

5 Considering Practical Matters

Most of the results presented above are of theoretical nature, yet we believe that GRASP can be efficient in practice. So far, we have assumed that only one routing operation was processed at a time, Section 5.1 provides an algorithm to handle concurrent routing operations as this will happen in practice. Second, Section 5.2 provides a sleep-wakeup scheme which improves the system lifetime.

5.1 Handling Concurrent Routing

So far, only one sensing request at a time was considered. This simplifying assumption allowed us to derive formal proof of convergence of the network topology but might be restrictive. In this section, we consider concurrent routing operations. GRASP should keep ensuring that *(i)* routing operations eventually succeed and *(ii)* two nodes should not get closer to each other than a fixed repulsion radius. We propose a set of modifications to GRASP so that such properties are still ensured and discuss their efficiency with respect to delays and distance covered.

Priority queue. We assume that each node maintains a queue of messages ordered by priority. To each message is associated a *Time from Emission* (TFE), updated every time a message is inserted or extracted from the priority queue. The message with the highest TFE being the head. The node movement is driven by the message being processed, namely the queue's head. Note that the number of older messages than a given message is finite and decreasing. This ensures that the delivery of messages is eventually guaranteed regardless of the heuristic chosen to forward the message.

Opportunistic forwarding. The simplest forwarding heuristic is to choose a static node to forward to. A less conservative heuristic is to forward a message to a node which is closer to the destination and, if moving, gets closer to the destination. To this end, nodes exchange their speed vector, piggybacked in beacons and fully characterized by the node's position and the destination of the current message being processed. A third heuristic, ensuring the repulsion radius between any pair of nodes, consists in considering the case of two moving nodes running into each other. Under this heuristic, two nodes getting in contact should be repulsed from one another, in analogy to the billiard model [15]. To this end, the two nodes exchange their positions and their current head (destination and age). The node the closest to the destination of the oldest head takes over all the messages and start processing them. The second node merely stops. This ensures that the two nodes are Moving away from one another.

5.2 Sleep-Wakeup

As proved in Section 3 and demonstrated in Section 4, a network of more than 1.5 the optimal number of nodes running GRASP converges to a triangular like lattice. Interestingly enough, the triangular lattice is a subgraph of the optimal hexagonal one. Based on this remark, we propose a simple but yet powerful sleep-wakeup scheme leveraging the network topology to increase the system lifetime. The triangular lattice (Figure 6(a)) is the union of three hexagonal lattices (Figures 6(b)-(d)), each node of the triangular lattice belonging to exactly two of them. Our sleep-wakeup scheme can be described as follow:

Clustering: the first step is to detect the triangular lattice-based components, the sleep-wake up algorithm being executed independently in each one of them. We assume that each node maintains the identifier of the component it belongs to. A node is able to determine if it is located at the center of an hexagonal tile using its neighbor's coordinates. A node at the center of an hexagonal tile sets its component identifier to its own identifier and forwards it to its neighbors forming the hexagonal tile. Otherwise, the component identifier remains unset. Upon reception of such a message, a node updates its component identifier if it is still unset or lower than the one received. Only nodes located at the center of an hexagonal tile forward it to hexagonal tile vertices. Otherwise, the message is ignored. Eventually, the nodes inside a triangular-lattice based component share the same component identifier. The node whose identifier is the one of the component is the natural leader of the component. This clustering task is periodically executed.

Sleep-wake up periods: each triangular component of the network alternates between three hexagonal configurations in a round-robin manner. The leader (node L in Figure 6) is in charge of spreading the sleep messages to the centers of the hexagonal tiles of the current configuration. The effect of a sleep message is to put in sleep mode the node receiving it for a given period of time. Leveraging the geometry of the network, the spreading of sleep messages can be done optimally with respect to the number of packets sent.

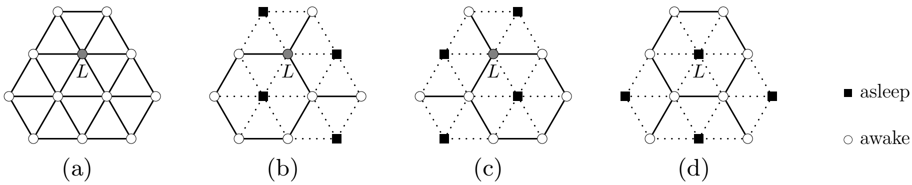


Fig. 6. Sleep-Wakeup scheme: (a) nodes inside a triangular lattice adopt in a cyclic manner three sleep-wakeup policies (b),(c) and (d). For the sake of readability, the leader L is reported in the three figures as a reference point. In each configuration, one third of the nodes are asleep and the communication graph is an hexagonal lattice.

Consider a system with the minimum required number of nodes deployed on an hexagonal lattice and assume that each node has a lifetime of one time unit, the system lifetime being also one time unit. Now, consider a system of $1.5N_{\text{opt}}$ nodes using the sleep-wakeup scheme presented above. Setting its working period (the time length during which the system stays in each of the three configurations) to 0.25 time unit, the global system lifetime is extended to 1.5 time units (i.e. six periods). Effectively, each node is asleep two periods out of six yielding a total number of awake periods of four, that is one time unit. In other words, the additional number of nodes over the optimal, namely 1.5, is fully leveraged by increasing the lifetime of the system up to 1.5.

6 Conclusions and Future Work

In this paper, we considered a network of mobile wireless sensors. Their mobility being controlled, we proposed Grasp, a novel and simple stateless algorithm which leverages nodes mobility to route sensing requests. Grasp transparently adapts to evolving region of interest, with respect to size or shape, without requiring to be explicitly aware of such changes. Our algorithm is independent from the communication medium and uses very simple forwarding and motion planning techniques. Thus it is directly applicable to any low capabilities wireless network. Assuming a disc model for communications and a random choice of the sensing locations inside the region of interest, we proved that a network running Grasp converges to a configuration ensuring greedy-connected coverage of the region. In that sense, the simplest routing algorithm leveraging nodes mobility provides the network with self-deployment properties for free. The number of nodes required to ensure convergence is 1.5 the optimal. Our simulation results matched our theoretical analysis. Finally, we provided Grasp with an ad-hoc sleep-wakeup scheme extending the system lifetime up to 50% without jeopardizing the greedy connected-coverage. This fully justifies the overhead factor to the optimal number of nodes. We also tackled concurrent requests routing. We plan to investigate these two tracks and evaluate Grasp behavior using a more realistic model for wireless communications.

References

1. Bai, X., et al.: Deploying wireless sensors to achieve both coverage and connectivity. In: Mobihoc 2006, pp. 131–142 (2006)
2. Tan, G., Jarvis, S.A., Kermarrec, A.M.: Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks. In: ICDCS 2008 (2008)
3. Butler, Z., Rus, D.: Controlling mobile sensors for monitoring events with coverage constraints. In: ICRA 2004, 26-May 1, 2004, vol. 2, pp. 1568–1573 (2004)
4. Jain, S., Fall, K., Patra, R.: Routing in a delay tolerant network. In: SIGCOMM 2004, pp. 145–158 (2004)
5. Mauve, M., Widmer, J., Hartenstein, H.: A Survey on Position-Based Routing in Mobile Ad-Hoc Networks. IEEE Network Magazine 15(6), 30–39 (2001)
6. Hull, B., et al.: Cartel: a distributed mobile sensor computing system. In: SenSys 2006, pp. 125–138 (2006)
7. Zhao, W., Ammar, M.H., Zegura, E.W.: A message ferrying approach for data delivery in sparse mobile ad hoc networks. In: Mobihoc 2004, pp. 187–198 (2004)
8. Zhao, W., Ammar, M., Zegura, E.: Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In: INFOCOM 2005, vol. 2, pp. 1407–1418 (2005)
9. Burns, B., Brock, O., Levine, B.N.: Mora routing and capacity building in disruption-tolerant networks. Ad-Hoc Networks 6(4), 600–620 (2008)
10. Huguenin, K., Kermarrec, A.M., Fleury, E.: Route in Mobile WSN and Get Self-Deployment for Free. Research Report 6819, INRIA (January 2009), `http://hal.inria.fr/inria-00357240/en/`
11. Kwon et al.: Resilient localization for sensor networks in outdoor environments. In: ICDCS 2005, pp., 643–652 (2005)
12. Fang, Q., Gao, J., Guibas, L.: Locating and bypassing routing holes in sensor networks. In: INFOCOM 2004, vol. 4, pp. 2458–2468 (2004)
13. Karp, B., Kung, H.T.: GPSR: greedy perimeter stateless routing for wireless networks. In: MobiCom 2000, pp. 243–254 (2000)
14. Iyengar, R., Kar, K., Banerjee, S.: Low-coordination topologies for redundancy in sensor networks. In: Mobihoc 2005, pp. 332–342 (2005)
15. Lubachevsky, et al.: Spontaneous patterns in disk packings. In: Bridges 1998: Conf. on Mathematical Connections in Art, Music, and Science (1998)