Servicing Wireless Sensor Networks by Mobile Robots

Xu Li, Inria Lille — Nord Europe

Rafael Falcon, Amiya Nayak, and Ivan Stojmenovic, University of Ottawa

ABSTRACT

Wireless sensor and robot networks (WSRNs) have emerged as a paradigmatic class of cyber-physical systems with cooperating objects. Due to the robots' potential to unleash a wider set of networking means and thus augment the network performance, WSRNs have rapidly become a hot research area. In this article, we elaborate on WSRNs from two unique standpoints: robot task allocation and robot task fulfillment. The former deals with robots cooperatively deciding on the set of tasks to be individually carried out to achieve a desired goal; the latter enables robots to fulfill the assigned tasks through intelligent mobility scheduling.

INTRODUCTION

The advent of microelectronics and advances in communications have made it technologically feasible and economically viable to develop low-power devices, *sensors*, that integrate general-purpose computing with multi-purpose sensing and wireless communications capabilities. It is expected that wireless sensor networks (WSNs) will have a significant impact on a wide array of applications ranging from military to civilian, establishing a ubiquitous computing environment that will pervade our society, thus redefining the way in which we live and work. Recently, in the attempt to integrate WSNs into the fabric of human activities, it was recognized that serving sensor networks by *mobile robots* (e.g., unmanned aerial vehicles) would bring forth superior performance. Robots are able to make complex decisions and take appropriate actions on themselves (e.g., controlled movement), sensor nodes (e.g., battery recharge), and the physical world (e.g., putting out a fire). They communicate directly or via multihop sensor paths.

Wireless sensor and robot networks (WSRNs) are the confluence of two traditional fields: WSNs and multirobot systems [1]. Leveraged by the control and mobility of robots, the networking process embraces a whole new set of possibilities. For example, robots may deploy and relocate sensors to improve network coverage, build routes and fix network partitions to ensure data communication, and change network topology to shape routing patterns and balance energy consumption. There is no limit to the benefits stemming from this mutual collaboration. Figure 1 illustrates a WSRN environment for long-term wildfire monitoring, an application example widely recognized in the literature (e.g., [2]).

Broadly speaking, the distinctive challenges posed by mobile robots in WSNs are *robot task allocation* and *robot task fulfillment*. In the sequel, we give a detailed discussion on both problems. We first elaborate on task allocation and display several solutions with various design considerations. Robot task fulfillment is addressed afterward, from three different perspectives: topology control, data collection, and sensor localization. We present a few of the best-known algorithms. Together, these sections offer a state-of-the-art picture of scientific research on servicing sensor networks by mobile robots. Finally, we discuss future research directions in this very promising field.

ROBOT TASK ALLOCATION

A prominent manifestation of inter-robot coordination that has drawn significant attention from the research community is multirobot task allocation (MRTA, a.k.a. task assignment) problem. A task is an atomic unit of responsibility that can be carried out independent from other tasks to achieve the ultimate system goal. Simply put, given a set of m tasks and n robots, we try to determine which robot should execute which task to optimize the overall system performance, which is a function of the individual *utility* of each robot. The utility measure is strictly task- and robot-dependent, and takes into account the profit gained after task completion as well as the cost of carrying out the task.

In the context of a WSRN, a *task* is an action required upon the occurrence of an event originating in the sensor field (e.g., target detection or sensor failure), and its *allocation* is the coordinated response of the robot team to such an event. Sensors transmit (usually via multihop routes) relevant event information (location, intensity, etc.) to one or more robots. Since the former are static and the latter mobile, finding a robot to which to report is a fundamental problem in WSRNs.

Although this problem lies beyond the scope of this short overview, we recommend a promis-

Centralized implementations are generally more accurate than localized schemes. However, they entail a non-negligible communication overhead and have poor fault tolerance and scalability properties. That is why distributed/localized algorithmic formulations seem to be gaining momentum nowadays.

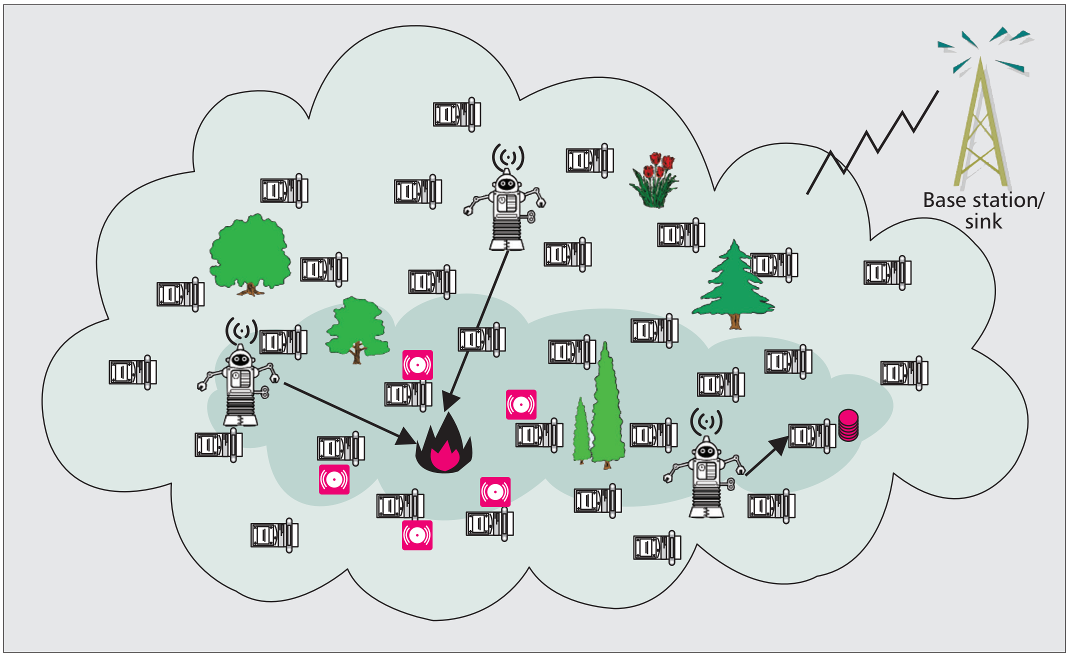

Figure 1. *A WSRN deployed for wildfire monitoring. Two firefighter robots (e.g., unmanned aerial vehicles) move to extinguish a detected fire, while one battery recharger robot relocates to deliver service to a sensor node upon request.*

ing solution in [3]. Sensors report to a nearby robot which is discovered by a truncated mesh that distributedly stores the robots' latest location. This structure is built by robots in a localized way through directional transmission of a location message in four geographic directions, with distance-based blocking applied at nodes where a horizontal transmission path and a vertical one meet. The directional transmission is realized by a well-known face-based geographic routing algorithm, without requiring directional antennas at individual sensors.

MRTA Problems: A Taxonomy

A domain-independent classification of task allocation problems in multirobot systems is unveiled in [4] as follows:

- Single-task (ST) vs. multitask (MT) *robots*: Indicates whether a robot can execute a single task at a time or multiple tasksconcurrently
- Single-robot (SR) vs. multirobot (MR) *tasks*: Indicates that a task requires exactly one robot or multiple robots to perform it
- Instantaneous (IA) vs. time-extended (TA) *assignments*: Indicates whether the available information on robots, tasks, and the environment allows only for an immediate allocation of tasks to robots or information on future tasks is known in advance and thus proper scheduling for task allocation can be arranged

A particular MRTA instance can be characterized by a combination of the above three axes. Table 1 lists the most frequently encountered MRTA cases in real-world WSRNs. It does not include MT-related combinations because MT robots (e.g., sampling a suspicious region while replenishing a sensor's battery) are not frequent in current WSRN literature.

Characterization of MRTA Architectures

Among the MRTA settings described in the above taxonomy, ST-SR-IA is the most studied in the literature because of its conceptual simplicity and wide applicability. The existing MRTA architectures can be grouped around two distinctive criteria: their *decision-making scope* and *model of cooperation*.

From the decision-making standpoint, MRTA approaches are either *centralized* or *distributed*. In a centralized protocol, decisions on robot task allocations are made by a single entity (e.g., a coordinator robot or the base station) once all the relevant information has been collected. Data flow from every entity (robot) to the coordinator, which runs a centralized algorithm (e.g., [5]) and notifies each robot on its set of assigned tasks. A distributed algorithm, on the other hand, bears a much higher degree of autonomy, since multiple entities are involved in deciding on an appropriate response. They can use knowledge pertaining to their immediate vicinities (e.g., one-hop neighboring robots) or, in the extreme case, rely on network-wide information. When decision making is based on limited knowledge, we are in the presence of a *localized* algorithm.

Centralized implementations are generally more accurate (i.e., yield high-quality, even optimal, solutions) than localized schemes. However, they entail non-negligible communication overhead, and have poor fault tolerance and scalability properties. That is why distributed/localized algorithmic formulations seem to be gaining momentum nowadays.

As to the model of cooperation (i.e., the underlying mechanism for allocating tasks to robots) used in current MRTA protocols, we distinguish between *minimalist* and *intentional* cooperation. In the minimalist model, a robot derives its individual action from the actions of other agents learned through explicit interaction, but there is no bargaining among individuals for task assignments. For instance, in *behavior-based* algorithms [6], a set of (often complementary) patterns an individual robot can execute must be defined. Such patterns (behaviors) are periodically communicated to teammates, and they trigger adaptive action selection over the group of agents.

Intentional cooperation exploits communication more heavily, as corporate actions are "negotiated" via interrobot message passing. This model of cooperation (*negotiation-based*) has enjoyed greater popularity in robotics research, probably due to its more intuitive formulation. Among its manifold implementations, *market-based* solutions [7] seem to be the current trend in WSRNs. In this economically inspired paradigm, robots work as free agents trying to maximize their own profit (utility). Tasks are assigned to robots by means of auctions. Normally, one of the robots performs as *auctioneer* and propagates the task to a bunch of *bidder* agents. Each bidder replies with its own profit value, and the task is assigned to the best bidder(s).

Name	Example
ST-SR-IA	A string of small fires are suddenly detected in a forest. Each can be put out by a single firefighter robot. Multiple robots pause their current task and coordinate impromptu to individually and immediately tackle the fires.
ST-SR-TA	As foreseen by the spectral analysis of the forest's heat map, several concurrent low-intensity fires broke out. The robots agree on a common schedule to solely focus on individually extinguishing them.
ST-MR-IA	A large-proportion, unexpected fire requires the undivided attention and combined effort of multiple robots.
ST-MR-TA	The robot team follows an a priori schedule to quench a series of fires of large intensity, some demanding the intervention of multiple, fully-concentrated team members.

Table 1. *Task allocation scenarios relevant to WSRNs.*

An Example: Auction Aggregation Protocol

Let us consider a recent market-based protocol [8] for the ST-SR-IA scenario in WSRNs. Upon occurrence of an event in the monitoring region, the robot receiving the information directly from the sensors (R3 in Fig. 2) becomes the auctioneer. The goal is to *find the fittest robot that can react to the event*. In a simple auction algorithm, R_3 would broadcast the task information (location, response time, etc.) to a group of bidder robots. Each bidder replies back to the auctioneer through a separate routing path. The auctioneer will then decide who is the winner and notify all bidders about it. The drawbacks here are the delay in selecting the best agent and the high communication overhead for large robot networks.

To simplify the process, [8] exploits some ideas from classical leader election algorithms in a tree. The auctioneer constructs a *response tree* rooted at itself. Each robot receiving the task offer includes the identifier of its parent robot (the one from which it first received the packet) in the message if forwarding is necessary. Those who do not forward the message (because all their neighbors already have it) or are not listed as parents by any neighbor declare themselves leaves and immediately bid to their parents. A parent node waits to hear from all its children, then *aggregates* their bids by only reporting the best one to its parent. The aggregation process is repeated along the response tree until the auctioneer selects the best network-wide bid and notifies the winning robot. This protocol is lighter in terms of communication costs than the previous version and may improve the response time too, depending on the robot network topology.

Toward Intelligent Task Allocations

Often, in multirobot task-driven scenarios it is not enough to identify which agents can indeed execute a given task, but intelligent task subdivision and subsequent agent reallocation becomes a must. Say that a fleet of aerial and ground autonomous vehicles have agreed to work together on putting out a threatening fire. Given the coordinates of the entire blazing region, for what section should each vehicle be responsible? The degree of overlapping between neighboring sections shall be minimized, and the specific limitations and resources of the vehicles (terrain/aerial access, extinguisher/cleaning materials, etc.) must be taken into account.

A negotiation algorithm that simultaneously splits and reallocates multirobot tasks was recently put forward in [9]. The task definition is generic enough to allow expressing different problems. The authors expand the famous "alternate offers" negotiation protocol in the theory of bargaining, which was originally formulated as a bilateral procedure, to the multilateral case. The basic idea is to let two entities negotiate by taking turns in sharing offers until an agreement is sealed. If one party is not satisfied with the offer, it will propose a counteroffer.

The negotiation process has two main components: the *protocol* level and the *generation* level. The former deals with the desired target reward per agent and the impatience to reach an agreement. The longer it takes for the agents to corporately arrive at a consensus, the less their allocated shares are worth. The generation level searches the space, aided by an evolutionary strategy, for a good share given a foreign proposal and considering its own resources, parameters, and limitations. A schematic representation of the proposed protocol is given in Fig. 3. It was tested across several domains with satisfactory results, although more research is needed concerning the estimation of the adversarial impatience.

Robot Task Fulfillment

After being assigned a task, a robot needs to geographically relocate itself to fulfill the task, i.e. to include the service consumer in its service range and deliver the required service. If the task is very specific, tied with a unique location, e.g. repairing a particular sensor node, the robot will just need to move directly toward that location. If the task is otherwise described in a general form for a large geographic region, e.g. fixing faulty sensors in the region, then the robot has to carefully plan its trajectory in order to guarantee service delivery as well as to satisfy task-specific requirements, mostly concerning service delivery latency.

Central to robot task fulfillment is thus the problem of robot *mobility scheduling*. Because sensors are severely resource-constrained, robots' actions while servicing sensors must impose minimal, if not zero, resource demand on the sensors. To give a conceptual idea of what this problem looks like and its polymorphic nature, we consider below three important scenarios: topology control, data collection, and sensor localization. We refer to the region assigned to a robot for service as the region of service (ROS). Multiple robots may be assigned to the same ROS.

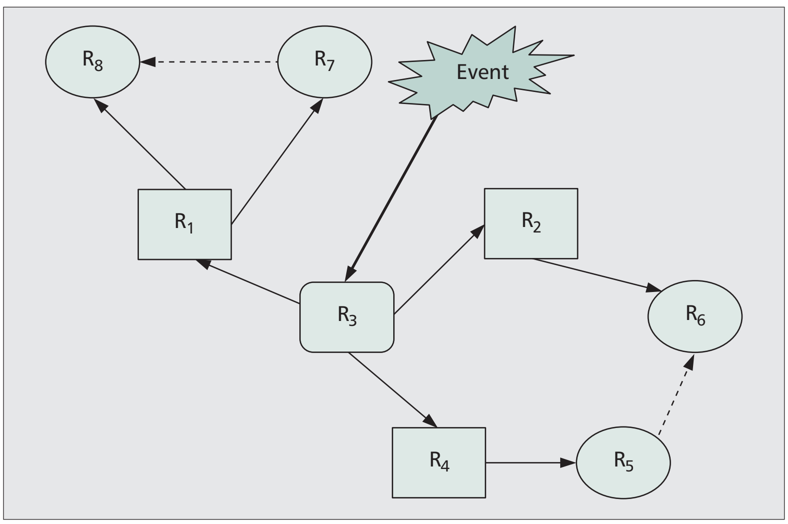

Figure 2. *Example of a response tree in [8]. Rectangles are parent nodes, circles are leaf nodes, and R_3 is the auctioneer. Dashed arrows are forwarded messages that did not create a parenting edge in the tree.*

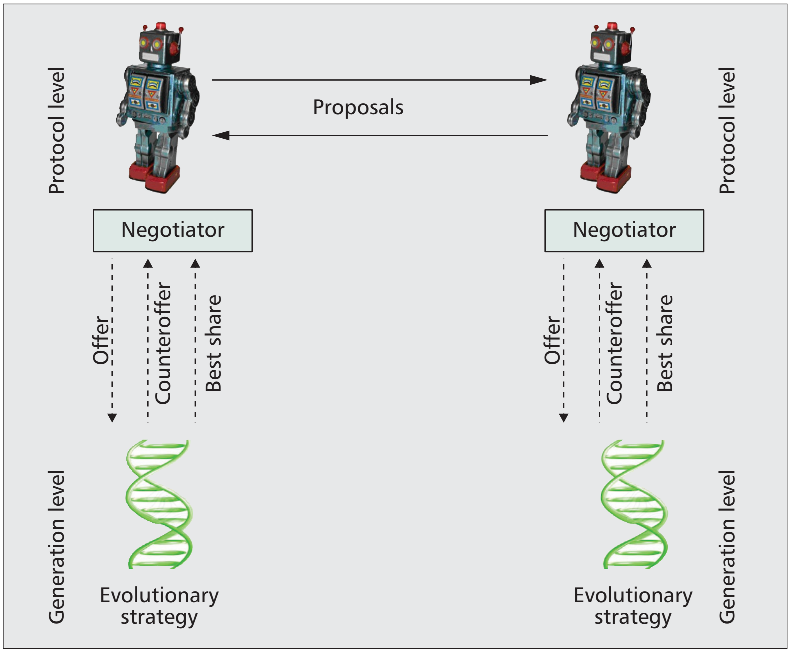

Figure 3. *Schematic representation of the negotiation protocol in [9].*

Topology Control

Robots may carry sensors as payload and move to explore the ROS. While traveling, they deploy sensors at proper positions to establish a network with a desired topology in terms of coverage and/or connectivity. If the ROS is bounded, the sensor deployment problem boils down to a graph traversal problem, where a virtual geographic graph is precomputed within the ROS according to the topology requirement. Each robot drops a sensor at visited empty vertices. A straightforward solution is depth first traversal. The challenge is to find an efficient traversal algorithm that minimizes the total moving distance and thus the deployment latency. Because robots possibly start at different locations, their deployed sensors may not finally interconnect unless they carry sufficient sensors to cover the entire ROS, or they are allowed to reload sensors. If the ROS is not bounded, it will be another challenge to ensure a network with good compactness. Compactness can be measured by the radius of maximum hole-free disc in the final network. It reflects the omnisensibility of the sensor network.

There are only a few such robot-assisted sensor placement algorithms. They (e.g. [10, 11]) assume known ROS boundaries and unlimited robot cargo capacity. Recently, the authors in [12] addressed the FOCUSED coverage (F-coverage) problem without using these assumptions. In F-coverage, sensors are required to surround a coverage focus, called the point of interest (POI), and maximize the coverage radius, that is, the minimal distance from the POI to uncovered areas. A localized Carrier-Based Coverage Augmentation (CBCA) protocol was proposed to incrementally construct a biconnected network with optimal F-coverage. Biconnectivity implies that there are at least two disjoint paths between every pair of nodes. The protocol relies on a triangle tessellation (TT) graph that is locally computable provided the POI coordinates are given and nodes agree on a common orientation (e.g., North).

In CBCA, robots enter the ROS from fixed locations, called *base points*, and advance straight to the POI. As soon as they get in touch with already deployed sensors, they search by multi-hop communication along the network border for the best sensor placement points (i.e., empty TT vertices) with respect to F-coverage optimization, and move to drop sensors at the discovered locations. Border nodes store locations of failed sensors inside the network as well as adjacent available deployment spots outside the network, and recommend them to robots during the search stage. This process is repeated until robots run out of sensors. Robots then return to base points for sensor reloading and re-enter the environment afterwards to augment existing F-coverage, as shown in Fig. 4a. Because robots move at different speeds, they may exchange their targets when in contact so as to minimize coverage augmentation delay. Techniques were developed to prevent target (sensor deployment point) contention (sensors recommending the same target to multiple robots) and resolve robot collision (multiple robots serving the same target).

Robots may also be used for improving the topology of an existing sensor network in addition to constructing a satisfactory one from scratch. Due to random dropping, some sensors are very likely to appear structurally redundant from a local perspective, while there are topologically vulnerable points (in terms of coverage or communication) in other parts of the network. The mobile robots could employ the redundant sensors for topology control (e.g., by conveying them to reported spots) subject to the limitation of cargo capacity on individual robots. The problem is then to find optimal robot tours for sensor pickup and delivery such that total travel distance (for energy efficiency) or maximum per robot travel distance (for delay efficiency) is minimized.

This problem can be modeled as a variant of the vehicle routing problem, which bears NP-hard complexity, and solved in a centralized fashion using exact methods (for small instances) and approximated algorithms (for large instances). Efficient population-based metaheuristics like genetic algorithms or ant colony optimization could be applied due to their concurrent exploration of the search space. The preferred approach is the one that returns the highest-quality solutions and remains fairly robust as the network size grows. The centralized solution requires full knowledge of redundant sensors and vulnerable points, which is expensive to obtain in resource-constrained sensor networks. Finding localized solutions using limited knowledge is an interesting open problem.

Data Collection

In a wireless sensor network (WSN), sensory data are to be collected at data sinks for processing and utilization. The main sensor-to-sink communication pattern is multihop message relay, as sinks are out of the transmission range of most of the sensors. The reporting paths from sensors to a sink form a reverse multicast tree rooted at the sink, with sensors in its vicinity enduring quick battery depletion due to their intensive message forwarding job. Nonuniform energy consumption degrades the network performance and shortens its lifetime. In delay-tolerant WSN applications like habitat and water quality monitoring, mobile robots may act as sinks to collect data and thus overcome this problem. As the robots move, the role of "hot spot" (i.e., heavily loaded nodes around sinks) rotates among sensors, resulting in more even energy distribution.

To maximize energy savings for sensors, *direct data collection* is the best option. That is, robots visit all sensors and obtain data directly from them. The service range of a robot is thus equal to the minimum of its own communication range and the communication range of the individual sensor it is servicing. This method entirely eliminates the message relay overhead of sensors and optimizes their battery levels. Direct data collection is often modeled after a traveling salesman

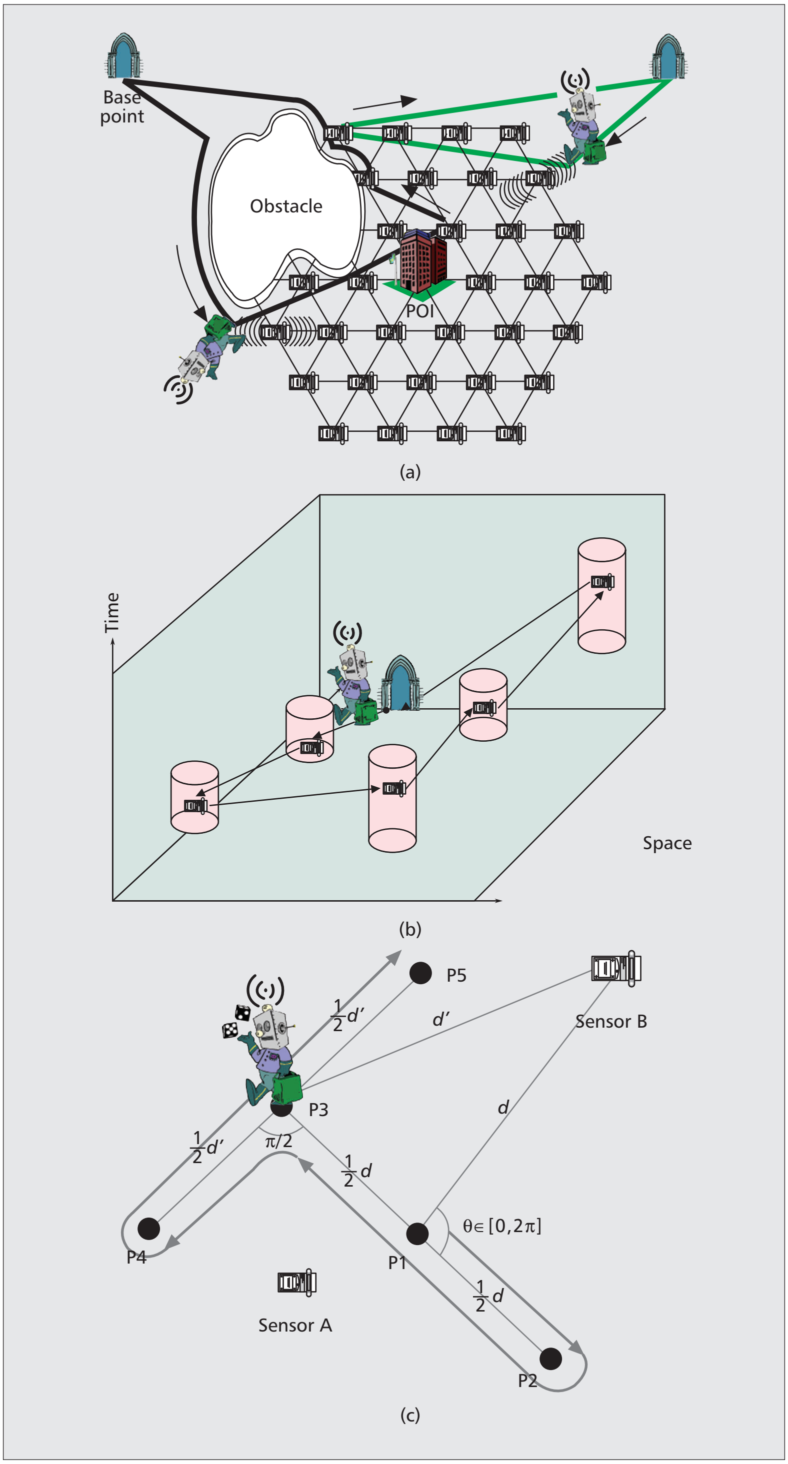

Figure 4. *Three instances of robot task fulfillment: topology control, data collection, and sensor localization: a) two robots, each loaded with a single sensor, enter the environment from two base points. They go to suggested sensor placement spots when reaching the network border and return to their base points for reloading after finishing the sensor placement; b) one sensor is selected in each event region, represented by a cylinder in a three-dimensional environment of space-time points. A robot travels along a shortest tour of these selected sensors to collect data and returns to its starting position; c) an iteration of robot heuristic movement toward sensor* B*, recommended by sensor* A*. The robot starts from* P_1 *and stops at one of* P_2*,* P_3*, and* P_5 *that is closer to* B *than* P_1*. The movement track and points* $P_2 \ldots P_5$ *are defined by its initial distance d to* B*, random choice of angle q at* P_1 *and random choice between* P_4 *and* P_5 *at* P_3*. The robot engages heuristic movement iteratively, with the endpoint of the current iteration being the start point of the next one, until* B *becomes localized.*

The advantages of using robots for sensor localization lie in reduced deployment cost (only a few beacon nodes are required) and communication overhead (only local communication is involved).

problem (TSP) variant or relaxed TSP, with constraints such as speed limit, time windows (due to buffer overflow), and event differentiation.

Recently, Xu *et al.* [13] suggested to select only a subset of the sensors to visit so as to reduce data collection delay while ensuring proper data gathering. They model each event as a 3D cylindric region. The underside of the cylinder is the region where the event is detected in space; the height is the event duration in time. The data collection problem is then solved in two steps. First, exactly one sensor is selected from each event region so that the minimum residual energy among all sensors is maximized after data gathering. The sensor selection is portrayed as a network flow problem, which runs in polynomial time irrespective of the robot's speed limit. Afterward, a TSP tour of the chosen sensors is computed by a heuristic algorithm to minimize robot speed requirement. This two-step algorithm requires complete knowledge of event regions, which is not available in reality. Therefore, the robot has to predict the event region based on past event data and adjust the selection of sensors and its tour iteratively, rather than by one-time computation. Figure 4b depicts a robot tour by this algorithm in a space-time setting.

Direct data collection is appealing in terms of sensor energy savings but significantly rises the collection delay because robots might advance slowly. Rendezvous-based data collection is thus investigated to achieve a trade-off between energy and time efficiency. Sensors transmit their measurements to a group of peers called rendezvous points (RPs) via multihop routes; a robot wanders around the network and retrieves data from encountered RPs. The use of RPs enables the robot to collect a large volume of data at a time without traveling a long distance and greatly decreases the collection delay. The direct-contact-based algorithm [13] may be extended to a rendezvous-based variant, where the sensors selected to be visited act as RPs and collect data from unselected ones.

Xing *et al.* [14] addressed the issue of balancing energy expenses and communication delay in rendezvous-based data collection, by jointly optimizing the selection of RPs, robot trajectory, and data transmission routes. They presented a heuristic algorithm based on the Steiner minimum tree (SMT). An SMT of data sources is built with an arbitrary data source as root. It has minimum length and thus energy optimal properties for transmission. In this tree, internal nodes are either existing data sources or Steiner points. Steiner points are added points for decreasing the total length of connection. As SMT is a lower bound of the optimal TSP tour of the sources, the algorithm selects RPs along it by a pre-order tree walk up to half of the maximum distance that the robot can travel within a given data collection time window.

Sensor Localization

In many real-world sensor network applications, a reported event (e.g. a fire in a woody area, an enemy tank in a battlefield or a survivor of a natural disaster) is meaningful and triggers a response only when it its location is known. Localization is an important problem dealing with how a sensor determines its spatial coordinates (position). A simple approach is to equip each sensor with a GPS (Global Positioning System) receiver that provides an accurate-enough location estimate. But this is not a cost-effective solution because a WSN is normally composed of a massive number of sensors and thus requires a significant financial investment. Moreover, GPS mainly works outdoors with no obstruction to satellite signal. This limits its applicability to indoor environments and has sparked a quest for alternative localization methods.

Non-GPS-based localization often requires certain location-aware devices, called *localization beacons*, which periodically emit beacon signals containing their spatial coordinates. Sensors infer their positions out of the spatial relations to the beacons in range, which are in turn assessed by using various signal features such as RSS (Received Signal Strength), ToA (Time of Arrival) and so on. To restrict positioning possibilities in a minimal area (i.e. to increase localization accuracy), they should be in touch with enough beacons. The number of beacons and their distribution thus have direct impact on the localization performance. A large number of uniformly distributed beacons will lead to better performance than a few crowdedly or linearly deployed ones.

Localized sensors may become new beacons and help other sensors self-localize. This iterative method reduces the initial number of beacons required but brings about aggregated localization error. Under these circumstances, mobile robots are introduced as an alternative budget-saving technique in delay-tolerant scenarios. The idea is the following: robots are aware of their own location; they transmit beacon signals conveying their up-to-date coordinates as they travel through the sensory field. Such beacon transmission locations correspond to the traditionally fixed beacons; on an individual sensor, a localization procedure is engaged during robot visit.

The advantages of using robots for sensor localization lie in reduced deployment cost (only a few beacon nodes are required) and communication overhead (only local communication is involved). Because sensors can be localized just when the robots are within their communication range and after they receive sufficient beacon signals from the robots, these advantages come at the expense of increased localization delay. Hence, robot trajectory has to be properly envisioned as optimal in length yet ensuring a quick, full, and accurate localization to every sensor. Because of random node dropping, sensor distribution is not known a priori, and robot trajectory should be planned on the fly rather than beforehand. Most existing algorithms overlook this trajectory planning problem and focused on the localization procedure. A few [15, 16] considered this problem, but simply adopted predetermined static paths or random mobility as a solution, thus providing no localization guarantee.

Li *et al.* [17] presented the first localized deterministic robot mobility scheduling algorithm with localization guarantee. This algorithm does not rely on any prior knowledge of the sensory field, and it works with noisy beacon signals. At the individual sensor level, it may accommodate any existing localization procedure as long as the procedure utilizes beacon signals for sensor localization. In this algorithm, each robot first visits a sensor by random movement, then performs a depth-first traversal (DFT) on the network graph under the currently visited sensor's instruction. That is, a visited sensor, after being localized, recommends an unvisited neighbor for the robot to visit next. Sensors run the built-in localization procedure to self-localize using received beacon signals.

DFT defines the visit sequence of a robot to sensors. The robot visit from a sensor to an unlocalized sensor in the sequence is implemented by heuristic movement following RSS-based distance measurement. Figure 4c illustrates how heuristic movement is engaged by a robot to visit sensor *B* after localizing sensor *A*. By DFT, each robot traverses a portion of the network and creates a traversal tree. Neighboring sensors that belong to different traversal trees negotiate on behalf of their localizing robots for the coordinate system to be used by these robots. The robot with the largest identifier wins; all other robots and their localized sensors are commanded through efficient tree-based flooding to align their coordinate systems and coordinates with the winner robot's. Eventually, all robots agree on a common coordinate system in which all sensors are localized. Topology control and node elimination are suggested to further shorten robot trajectory.

Future Challenges

Robot task allocation and robot task fulfillment have been largely treated in literature as two standalone scenarios. Recently, they were put together in the *robot dispatch problem* [18], in which sensors monitor the environment and report events occurring in the field; robots are then dispatched to visit these event locations (task allocation), for example, to conduct more advanced analysis or provide timely event response. As new events arise, multiple rounds of robot dispatch may be needed. The goal is to schedule the robot traveling paths (task fulfillment) in an energy-balanced way so that their overall lifetime (the number of dispatch rounds) is maximized. This combinatorial optimization problem is NP-hard [18]. A centralized suboptimal solution and its distributed implementation (where global information is gathered at a dynamically determined sensor node that hosts and runs the centralized scheme) were described in [18].

Designing distributed or localized solutions to the robot dispatch problem stands as a promising research avenue. For the single-event case, the service discovery algorithm iMesh [3] finds the nearest (or a nearby) robot with lower communication overhead than [8]. An alternative approach was discussed in [8]. It assumes that the event was reported already to one of the robots, without explaining how to do it, while iMesh in fact discusses how to report it but emphasizing on immediate briefing to the nearest robot. We may mix [3] and [8] into a single scheme, with report to a nearby robot, which can afterward consult other teammates in its neighborhood to actually detect the closest one. The quest for the most suitable robotic agent undertaken by iMesh could embrace, for instance, the *reluctance* of a robot to perform the task (which is inversely proportional to its remaining battery level) and employ *distance* × *reluctance* in place of *distance* alone as the selection metric.

A future research direction is to study how to expand this hybrid scheme (mixture of [3] and [8]) so that it can handle multiple events. One option is to apply the scheme iteratively and allow the robot who agreed to tend to the current event to reduce its energy level accordingly, thus pretending it is already at the event venue, before the agent responsible for the next event is appointed. Should two or more events be concurrently accepted, those still unvisited may be sorted in a different order than the already accepted ones. The alternative is to run a similar competition under the assumption that events do not have access to all robots, which is a desirable feature for a communication-efficient solution. Instead, for each event request, there is a dedicated robot (obtained by [3]) that runs an auction, and the winner robot responds to the request.

In previously introduced algorithms, communication among robots is not implemented directly. For example, in [8] it relies on the assumption of constant connectivity among robots and is realized by the distributed auctioning process in order to determine the most suitable bidder to tend the event in the sensor field. Explicit robot-robot communication without such a strong assumption may be needful in developing future distributed solutions for robots to service WSNs. Robots with limited communication range are very likely far apart from each other, thus not forming a connected network themselves, and they may not be aware of each other or each other's location due to robot addition/removal and mobility. Since sensors may relay messages for robots, the challenge is to dynamically establish and maintain a connected robot graph where edges are multihop routing paths composed of sensors. Preferably, the graph is a planar so that existing data communication protocols can be readily adopted. Both sensor and robot networks could benefit from maintaining their backbones in the form of connected dominating sets [19].

Robots with limited communication range are very likely far apart from each other, thus not forming a connected network themselves, and they may not be aware of each other or each other's location due to robot addition/removal and mobility.

Acknowledgments

This work was partially supported by NSERC CRDPJ 386874-09 "Reliable and secure QoS routing and transport protocols for mobile ad hoc networks" and NSERC Discovery grants.

Ivan Stojmenovic is also associated with EET, Faculty of Technical Sciences, University of Novi Sad, Serbia. This work was partially supported by grant "Innovative electronic components and systems based on inorganic and organic technologies embedded in consumer goods and products," TR32016, Serbian Ministry of Science and Education.

Since sensors may relay messages for robots, the challenge is to dynamically establish and maintain a connected robot graph where edges are multi-hop routing paths composed of sensors. Preferably, the graph is a planar so that existing data communication protocols can be readily adopted.

References

[1] S. L. Kim, W. Burgard, and D. Kim, "Wireless Communications in Networked Robotics," *IEEE Wireless Commun.*, vol. 16, no. 1, Feb. 2009, pp. 4–5.

[2] K. Akkaya, F. Senela, and B. McLaughlana, "Clustering of Wireless Sensor and Actor Networks Based on Sensor Distribution and Connectivity," *J. Parallel and Distrib. Comp.*, vol. 69, no. 6, June 2009, pp. 573–87.

[3] X. Li, N. Santoro, and I. Stojmenovic, "Localized Distance-Sensitive Service Discovery in Wireless Sensor and Actor Networks," *IEEE Trans. Computers*, vol. 58, no. 9, Sep. 2009, pp. 1275–88.

[4] B. Gerkey and M. Matariċ, "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," *Int'l. J. Robotics Research*, vol. 23, no. 9, Sep. 2004, pp. 939–54.

[5] S. Liu, T. Sun, and C. Hung, "Multi-Robot Task Allocation Based on Swarm Intelligence," *Multi-Robot Systems, Trends and Development*, T. Yasuda, Ed., InTech, 2011, pp. 393–408.

[6] L. E. Parker, "ALLIANCE: An Architecture for Fault-Tolerant Multi- Robot Cooperation," *IEEE Trans. Robotics and Automation*, vol. 14, no. 2, Apr. 1998, pp. 220–40.

[7] M. B. Dias *et al.*, "Market-Based Multirobot Coordination: A Survey and Analysis," *Proc. IEEE*, vol. 94, no. 7, July 2006, pp. 1257–70.

[8] V. Malbasa, I. Mezei, and I. Stojmenovic, "Communication Aspects of Coordination in Robot Wireless Sensor Networks," *IEEE Robotics and Automation Mag.*, vol. 17, no. 4, Dec. 2010, pp. 63–69.

[9] C. Rossi, L. Aldama, and A. Barrientos, "Simultaneous Task Subdivision and Allocation for Teams of Heterogeneous Robots," *Proc. IEEE ICRA*, 2009, pp. 946–51.

*[10] C. Y. Chang et al., "Obsta*cle- Resistant Deployment Algorithms for Wireless Sensor Networks," *IEEE Trans. Vehic. Tech.*, vol. 58, no. 6, Nov. 2009, pp. 2925–41.

[11] C. Y. Chang *et al.*, "An Obstacle- Free and Power-Efficient Deployment Algorithm for Wireless Sensor Networks," *IEEE Trans. Systems, Man, Cybernetics — Part A*, vol. 39, no. 4, Apr. 2009, pp. 795–806.

[12] R. Falcon, X. Li, and A. Nayak, "Carrier-based Focused Coverage Formation in Wireless Sensor and Robot Networks," *IEEE Trans. Automatic Control*, vol. 56, no. 10, Oct. 2011, pp. 2406–17.

[13] X. Xu, J. Luo, and Q. Zhang, "Delay Tolerant Event Collection in Sensor Networks with Mobile Sink," *Proc. IEEE INFOCOM*, 2010, pp. 2471–79.

[14] G. Xing *et al.*, "Rendezvous Design Algorithms for Wireless Sensor Networks with a Mobile Base Station," *Proc. ACM MobiHoc*, 2008, pp. 231–39.

[15] D. Koutsonikolas, S. M. Das, and Y. C. Hu, "Path Planning of Mobile Landmarks for Localization in Wireless Sensor Networks," *Comp. Commun.*, vol. 30, no. 13, Sep. 2007, pp. 2577–92.

[16] K. Kim and W. Lee, "MBAL: A Mobile Beacon-Assisted Localization Scheme for Wireless Sensor Networks," *Proc. ICCCN*, 2007, pp. 57–62.

[17] X. Li *et al.*, "Dynamic Beacon Mobility Scheduling for Sensor Localization," *IEEE Trans. Parallel and Distrib. Sys.*, to appear, 2012.

[18] Y. C. Wang, W. C. Peng, and Y. C. Tseng, "Energy-balanced Dispatch of Mobile Sensors in a Hybrid Wireless Sensor Network," *IEEE Trans. Parallel and Distrib. Sys.*, vol. 21, no. 12, Dec. 2010, pp. 1836–50.

[19] K. Islam and S. G. Akl, "A Local Algorithm to Compute Multiple Connected Dominating Sets in Wireless Sensor Networks," *Int'l. J. Parallel, Emergent and Distrib. Sys.*, vol. 26, no. 5, 2011, pp. 369–80.

Biographies

XU LI (xu.li@inria.fr) received a Ph.D. (2008) degree from Carleton University, Canada, an M.Sc. (2005) degree from the University of Ottawa, Canada and a B.Sc. (1998) degree from Jilin University, China, all in computer science. Prior to joining Inria, France, as a research scientist, he held post-doctoral fellow positions at the University of Waterloo, Canada, Inria/CNRS, France, and the University of Ottawa, Canada. He is on the editorial boards of *Transactions on Emerging Telecommunications Technologies*, and *Parallel and Distributed Computing and Networks*, and is/was a co-guest editor of a few international journals. He has been appointed a TPC Co-Chair for the Ad Hoc and Sensor Networking Symposium at IEEE GLOBECOM 2013. His current research focuses on machine-to-machine communications and mobile social networks. He has published more than 50 different works in these fields.

RAFAEL FALCON [S'09] (rfalcon@uottawa.ca) received a Ph.D. degree from the University of Ottawa, Canada, in 2012, and M.S. and B.S. (with highest honors) degrees from Universidad Central de Las Villas, Cuba, in 2003 and 2006, respectively, all in computer science. He was a visiting Scholar for Hasselt Universiteit, Belgium, and the Universidad de Granada, Spain. He has co-edited two Springer volumes on fuzzy and rough set theories and served as a reviewer for top-tier scientific journals. His current research interests embrace wireless sensor and robot networks, distributed computing, bio-inspired optimization, rough sets, fuzzy logic, and knowledge-based clustering. He is a member of the International Rough Set Society and the IEEE Computational Intelligence Society.

AMIYA NAYAK [SM'04] (anayak@eecs.uottawa.ca) received a B.Math. degree in computer science and combinatorics and optimization from the University of Waterloo in 1981 and a Ph.D. degree in systems and computer engineering from Carleton University in 1991. He has more than 17 years of industrial experience, working at CMC Electronics, Defense Research Establishment Ottawa (DREO), EER Systems, and Nortel Networks in software engineering, avionics and navigation systems, simulation, and system-level performance analysis. He is on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, *International Journal of Parallel, Emergent, and Distributed Systems*, *International Journal of Computers and Applications*, *International Journal of Computer Information Technology and Engineering*, and *International Journal of Computing and Information Science*. Currently, he is a full professor in the School of Electrical Engineering and Computer Science at the University of Ottawa. His research interests are in the areas of mobile ad hoc and sensor networks, fault tolerance, and distributed systems/algorithms.

IVAN STOJMENOVIC [F'08] (ivan@eecs.uottawa.ca) received his Ph.D. degree in mathematics. He has held regular and visiting positions in Serbia, Japan, the United States, Canada, France, Mexico, Spain, the United Kingdom (as Chair in Applied Computing at the University of Birmingham), Hong Kong, Brazil, Taiwan, and China, and is a full professor at the University of Ottawa, Canada, and an adjunct professor at the University of Novi Sad, Serbia. He has published over 300 papers, and edited seven books on wireless, ad hoc, sensor and actuator networks, and applied algorithms with Wiley. He is editor of over a dozen journals, Editor-in-Chief of *IEEE Transactions on Parallel and Distributed Systems* (since January 2010), and founder and Editor-in-Chief of three journals (*Journal of Multiple-Valued Logic and Soft Computing*, *International Journal of Parallel, Emergent, and Distributed Systems*, and *Ad Hoc & Sensor Wireless Networks*). He is one of about 250 computer science researchers with h-index of at least 50 and has over 11,000 citations. He received four best paper awards and the Fast Breaking Paper in October 2003 from Thomson ISI ESI. He is a recipient of the Royal Society Research Merit Award, United Kingdom. He received the Excellence in Research Award of the University of Ottawa in 2009. He has (co)chaired and/or organized over 100 workshops and conferences. More details can be found at http://www.site.uottawa.ca/~ivan.