

Exploiting Actuator Mobility for Energy-Efficient Data Collection in Delay-Tolerant Wireless Sensor Networks

Xu Li*, Amiya Nayak* and Ivan Stojmenovic*[†]

*SITE, University of Ottawa, Canada
{xuli,anayak}@site.uottawa.ca

*[†]EECE, University of Birmingham, UK
stojmenovic@storm.ca

Abstract

A wireless sensor network (WSN) is a large-scale ad hoc multi-hop network deployed (usually, at random) in a region of interest for surveillance purpose. One of its fundamental tasks is to gather sensor readings from the sensory field at data sinks. Research has shown that sensors near a sink deplete their battery power faster than those far apart due to the heavy overhead of relaying messages. Non-uniform energy consumption causes degraded network performance and shortened network lifetime. In this paper, we address the problem of energy-efficient data collection by mobile actuators in delay-tolerant WSN. We present a taxonomy and a comprehensive survey of state of the art on the topic.

1 Introduction

Wireless sensors are battery-powered sensing devices that are able to communicate through radio frequency. Oftentimes, they are densely dropped at random in a large region of interest for object monitoring and target tracking. Once deployed, they self-configure into a connected wireless network and operate completely on their own. To minimize deployment budget, sensors are intentionally made low in cost. They have very limited resources such as energy, memory, computation and communication capabilities. Yet, they are expected to operate for long-term use. Therefore, energy efficiency is of paramount importance in such a wireless sensor network (WSN).

In WSN, sensor readings are transmitted to one or more pre-defined data sinks for analysis and processing. The main sensor-to-sink communication pattern is multi-hop message relay, as most of sensors are out of the range of the sink. Research [18, 20] has shown that, the closer to a sink a sensor is, the faster its battery exhausts. It is because, com-

pared with sensors far apart from a sink, nearby sensors are shared by more data communication paths to the sink and have heavier message relay load. According to [17, 30], by the time when the one-hop neighboring sensors of a sink deplete their battery power, those farther away may still have more than 90% of their initial energy.

Uneven energy depletion leads to degraded network performance and shortened network lifetime. If sensors around a sink all run out of energy, the sink will be isolated from the network; if all sinks are isolated, then entire network fails. Since periodical manual replacement/recharge of sensor batteries is often infeasible due to operational factors, it is desired to minimize and balance energy usage among sensors for network lifetime elongation purpose.

Power-aware routing [4, 23, 24] have been studied to avoid energy-scarce sensors and achieve network lifetime improvement. As indicated in [14, 20, 24], proper use of multi-level transmission radii can also help save message relay and balance energy consumption. It was as well suggested to use non-uniform node distribution (i.e., the closer an area is to a sink, the higher node density it has) to mitigate message relay load and increase network lifetime [17, 25, 31]. The first two approaches have limited effectiveness since sensors around a sink are very likely to be critical to sink connectivity and can not be skipped, while the third approach reduces network sensing coverage, which is the functional basis of any sensor network.

Recently, it was noticed that mobile data-collection actuators (also called mobile sinks [3, 8, 19, 29], mobile routers [11], mobile elements [7, 32], or data MULEs [22, 27]) can be used to improve network lifetime without bringing above-mentioned negative impacts on the network. In this paper, we draw attention to this research subject with a particular interest in delay-tolerant WSN. We present a taxonomy in Section 2 and review the literature in Section 3 and 4. Finally, we conclude the paper in Section 5.

2 Energy efficiency by actuator mobility

Actuator mobility may be classified as *uncontrollable* or *controllable* in general. The former is usually obtained by attaching an actuator node on some mobile entity such as animal or shuttle bus, which already exists in the deployment environment and is out of control of the network. The latter is achieved by intentionally adding a mobile entity e.g., mobile robot or unmanned aerial vehicle, into the network to carry the actuator node. In this case, the mobile entity is an integral part of the network and can be fully controlled by the network. Both uncontrollable mobility and controllable mobility can be exploited for energy efficiency.

In delay-tolerant WSN for applications like habitat monitoring and water quality monitoring, energy saving embraces a lot of possibilities. To maximize energy saving, direct-contact data collection is the best option. In this approach, a mobile actuator visits (possibly at slow speed) data sources and obtains data directly from them by one-hop communication. It will physically carry collected data to sink nodes if it itself is not sink. Because sensors do not need to forward messages for each other, this approach minimizes energy usage among sensors for communication at most. Its main concern is computation of best actuator tour that covers all data sources and minimizes travel distance.

Direct-contact data collection has great advantage in energy saving. But it generates large data collection delay because of the actuator's low moving speed. Under this circumstance, rendezvous-based data collection is brought into attention to achieve trade off of energy consumption and time delay. In this approach, data sources send data to a subset of sensors called *rendezvous points* (RPs) by multi-hop communication; an actuator moves around in the network and retrieves data from encountered RPs. The use of RPs enables the actuator to collect a large volume of data at a time without traveling a long distance and thus decreases

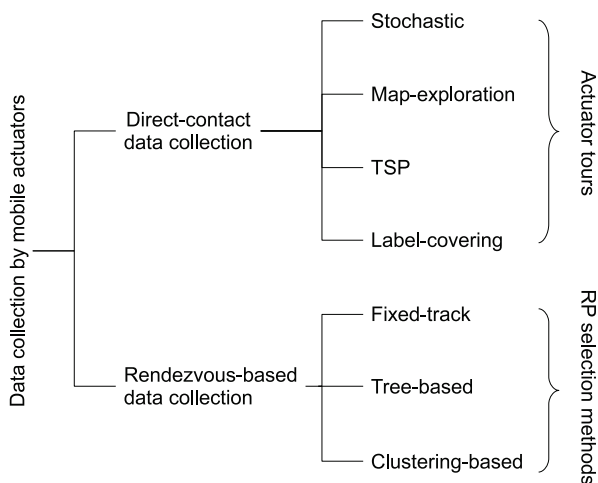


Figure 1. A taxonomy

data collection latency. A solution algorithm focuses mainly on how to select best RPs under different constraints.

Figure 1 depicts a taxonomy of mobile-actuator-based data collection methods. At the top level of the taxonomy are direct-contact and rendezvous-based approaches. The former is divided into four groups according to their employed actuator tours; the latter is partitioned into three classes by their employed RP selection methods.

3 Direct-contact data collection

3.1 Stochastic data collection trajectory

Shah et al. [22] considered stochastic actuator mobility and proposed a simple data collection algorithm. In the proposal, sensors buffer their measurements locally and wait for the arrival of an actuator. Each actuator moves randomly and collects data from encountered sensors. Because actuator trajectory is random, there is not performance guarantee on data collection delay.

In the case of stochastic actuator mobility, energy consumption at sensor side is only due to actuator discovery and subsequent data transfer. Assume that an actuator periodically broadcasts a beacon message while moving. It is very expensive to detect actuator arrival by keeping listening to the beacon message. Chakrabarti et al. [5] show that, if actuators (e.g., shuttle buses) move along regular path, then sensors can predict their arrival after being allowed a learning curve for their movement pattern.

According to [1], message loss probability decreases with increased sensor-to-actuator distance. If a sensor transmits as soon as it discovers an actuator, data may not be delivered or may be delivered with many retrials, wasting energy. Suppose the actuator passes by sensors along straight line. Anastasi et al. [2] proposed to start data transfer only in the time interval with minimum message loss probability, which is exactly around the minimum-distance point.

3.2 Map exploration for data collection

Li and Stojmenovic [16] considered controllable actuator mobility and presented a map exploration method for data collection in a bounded environment. The actuator is assumed to have a communication radius not smaller than sensor communication radius r_c . It is able to detect obstacles and boundaries of the environment. It does not know the locations of sensors, and aims to explore the entire environment and obtain data from encountered sensors.

To reduce search space, actuator trajectory is restricted to a virtual graph enclosed by the environment boundaries. The graph is constructed in such a way that the aggregation of all the discs of radius r_c centered at each vertex covers

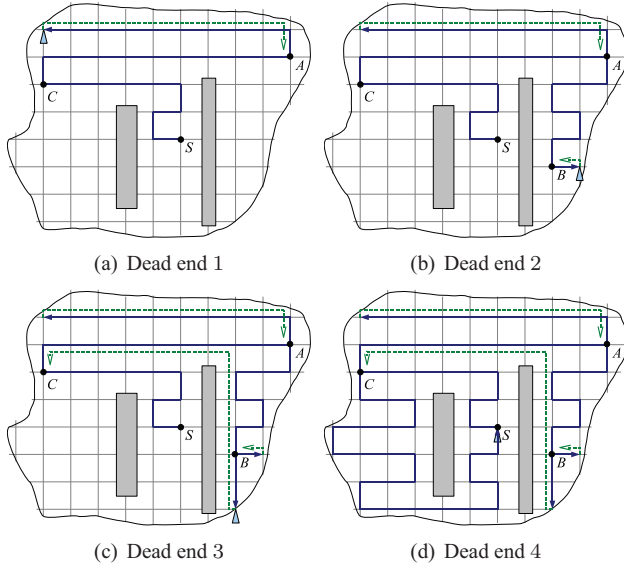


Figure 2. Snake-like traversal

the entire environment. Boundary effect is ignored for simplicity. A square grid tessellation of edge length $\sqrt{2}r_c$ or an equilateral triangle tessellation of edge length $\sqrt{3}r_c$ is suggested. By exploring such a graph, the actuator is guaranteed to be able to visit the communication range of every sensor and collect data from it. The actuator has sufficient memory to store the virtual graph. Now the data collection problem is converted to graph traversal problem. One suggested solution by the authors is depth first traversal (DFT) [28]. They also proposed two new traversal methods, *snake-like traversal* and *boundary traversal*.

In snake-like traversal, the four geographic directions, i.e., North, West, South, and East, are assigned distinct ranks, and the actuator moves step by step toward local open direction with highest rank. At each movement step, it travels a proper distance so that it stops exactly at a vertex in the virtual graph. At each visited vertex, it discovers nearby sensors and collects data from them. If the actuator's movement is obstructed (by an obstacle, or environment boundary, or visited vertex) at a step, it chooses to move to the direction of next highest rank. In the case of dead-end, i.e., when no obstructed direction exists, the actuator moves back to most recently visited vertex that is adjacent to an unvisited vertex, and resume exploration from there. In the course of backward movement, it travels along a shortest path composed of visited vertices in the graph so as to avoid obstacles.

Figure 2 illustrates the snake-like graph traversal method over a square grid tessellation that is constructed within an bounded area in potato shape. The area contains two obstacles, shown as rectangular blocks in the figure. The actuator starts from position S and travels in the order of preference $West > East > North > South$. Its trajectory is marked

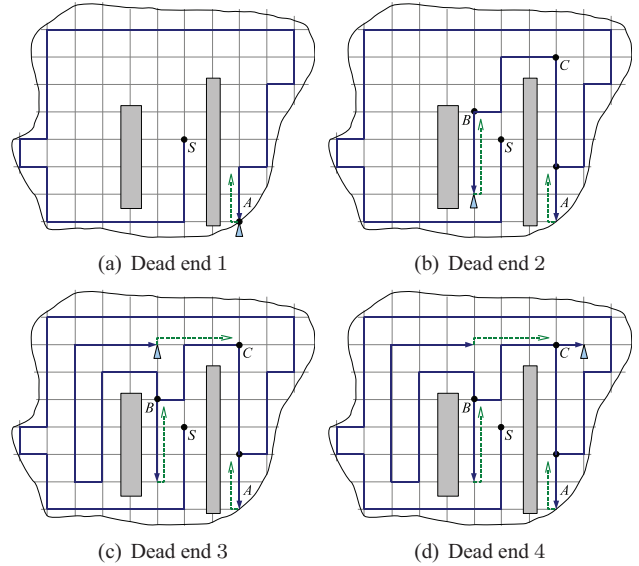


Figure 3. Boundary traversal

by arrowed lines; its current position is indicated by a small triangle. The four sub-figures show the four dead-end situations during the actuator's course of graph traversal. Broken arrowed lines show which vertex and along which path it moves back to in those situations. In Fig. 2(d), the actuator stops moving or starts another round of traversal since all vertices have been visited.

Boundary traversal simulates the process of potato peeling. The actuator first moves toward an arbitrary direction until it hits the boundary of the environment; then it travels along the boundary of the unknown part of the environment in one direction, i.e., either clockwise or counter-clockwise. Every time when the actuator makes a circle along the unknown area, a layer of unvisited vertices are "peeled" off. As the number of unvisited vertices keeps decreasing, the traversal process eventually terminates. The same recovery technique for dead-end situation is used as in snake-like traversal. Figure 3 illustrates this traversal method.

3.3 TSP tour for data collection

With controllable actuator mobility and knowledge of sensor locations, data collection delay can be reduced by properly selecting actuator trajectory. It is not difficult to conclude optimal actuator mobility scheduling is generally equivalent to the NP-complete Traveling Salesman Problem (TSP) [13]. Informally, the TSP problem is: given a number of cities (i.e., sensors), find the shortest tour that visits each city exactly once and returns to the starting city.

For data collection purpose, it is sufficient that an actuator visits the communication range (modeled as disc) of a sensor instead of the sensor itself. From this consideration, Nesamony et al. [19] formulated the actuator traveling

problem as a variant of TSP, known as Traveling Salesman with Neighborhood (TSPN), where an actuator is required to pass through the neighborhood of each sensor exactly once. The authors presented an algorithm for finding the best possible actuator tour. This algorithm first determines the visiting order of the communication discs (ranges) of sensors. In this process, some ordering constraints may apply. If there is not any constraint, then the most intuitive way is to order the discs based on the TSP order of their representative points. The representative point of a disc could be selected respect to the geometric information of the disc in the entire field. Then the algorithm computes a point set as TSP input. The point set initially contains the starting point of the actuator and the representative point from each disc. The points in this set are updated in TSP order. Each point is updated according to some policies with respect to its two neighboring points, and the updated point is used immediately for updating the next point. The actuator tour defined by the new point set is guaranteed to have smaller length than the old one. The point set is updated iteratively in this way until the length of the tour stabilizes.

Sensors have different storage capacities. They need to be visited according to their buffer limitation to avoid data loss. Gu et al. [7] addressed the impact of sensor buffer limitation on the TSP for actuator mobility and presented a partitioning-based scheduling (PBS) algorithm. In this algorithm, sensors are partitioned into groups, called *bins*, B_1, B_2, \dots . The buffer overflow times of sensors in B_i are in the same range; the range of buffer overflow times for bin B_{i+1} is twice that of bin B_i . Each bin is further geographically partitioned into sub-bins such that the sensors in the same sub-bin are close to each other. The actuator travels along a super-cycle composed of visit cycles of bins. Each visit cycle includes exactly one sub-bin from each bin in order, and it starts from the sensor with minimum buffer overflow time in a sub-bin of B_1 . In each visit cycle, a sub-bin in B_i is followed by a closest sub-bin in B_{i+1} . The actuator mobility scheduling is then reduced to the classic TSP problem in each sub-bin.

In the authors' continuous work [6], an improved version of PBS was presented to handle messages with different delivery deadline. Urgent messages do not wait for actuator's arrival; they are immediately relayed to nearby sensors within a pre-determined number of hops that are visited more frequently by the actuator. An iterative buffer overflow time reduction phase is used to ensure all nodes to be covered by the actuator tour without buffer overflow or delivery deadline violation.

3.4 Label-covering tour for data collection

Sugihara and Gupta [26, 27] addressed actuator path selection for minimizing data collection delay. They relaxed

the requirement for exact one-time visit of the actuator to each sensor's communication range. The intuition is that, the actuator's travel time could be long if the length of the intersection of its path and the communication range of each sensor is short. Exact one-time visit may not always be a winning strategy. On the contrary, multi-visits together with proper speed control may yield a better solution.

The authors simplified the path selection problem by reducing search space to a complete geographic graph, where there are vertices at sensors' locations. The actuator moves in this graph along edges from vertex to vertex. Each edge is associated with a cost and a set of labels. Cost is defined as Euclidean length of the edge; the label set represents the set of sensors whose communication ranges intersect with this edge. The objective is to find a shortest (minimum-cost) tour whose associated label set covers all sensors.

The authors proved that the shortest label-covering tour problem is NP-hard, and presented an approximation algorithm to solve it. The algorithm finds a TSP tour by any TSP solver. Then, by dynamic programming, it finds the shortest label-covering tour that can be obtained by applying short-cutting to the TSP tour.

4 Rendezvous-based data collection

4.1 RP selection along fixed track

Kansal et al. [11] proposed to use a straight-line actuator path for data collection. There is a single actuator in the network. It moves along a straight line and broadcasts a beacon while moving. A receiver node rebroadcasts the beacon if and only if the beacon comes along a shortest path it has seen. Then a number of minimum hop reporting trees are established along the actuator path. This tree construction process takes place only once.

The root of each reporting tree is a RP. Each sensor sends its measurements along an upward path to the root of its residing trees. When the actuator arrives in its neighborhood, a RP sends its own data together with the data received from its tree members to the actuator. Two motion control algorithms were presented to adjust the speed of the actuator to avoid data loss.

The authors considered multi-actuator scenarios in their continuous work [10]. The sensory field is divided into equal-sized areas, each with a single actuator. The single actuator algorithm is run in each area. A load balancing algorithm is executed by an elected actuator to ensure that each actuator path be assigned the same number of sensors.

Xing et al. [32] considered the case that an actuator moves along a fixed track of arbitrary shape. They assumed that data aggregation is applied at sensor nodes and that the total energy consumption for message transmission along a

multi-hop path is proportional to the Euclidean distance between sender and receiver. The objective is to select RPs along the actuator track such that the total length of edges that connect sources to RPs is minimized.

They presented a Minimum Spanning Tree (MST) [12] based algorithm. In this algorithm, an optimal set of MSTs that connect all sources to the actuator track is constructed in the Euclidean domain. The set is optimal in that the total length sum of its member MSTs is minimal. Note that each individual MST is rooted at a point on the track, and it does not always span all data sources. Since the MST set are approximations of the optimal reporting trees in practice for data gathering, the roots of the MSTs are taken as RPs.

4.2 RP selection along reporting tree

Xing et al. [33] studied RP selection along a geometric tree that approximates the reporting tree (rooted at a static base station) of data sources. RPs must be properly selected so that, the length of the actuator tour is not larger than the maximum distance that the actuator can travel within a given data collection deadline. Both constrained and unconstrained actuator mobility are considered.

A greedy algorithm was presented for actuator mobility constrained on the tree. Each tree edge is assigned a weight, equal to the number of sources in the subtree rooted at its upper end (the end toward the root). A sub-tree of total weight equal to half of the maximum travel distance is constructed by greedily selecting edges of maximum weight from the tree. A partial tree edge may be selected at last to ensure exact total weight. The actuator tour is then determined by pre-order traversal of this sub-tree.

In the case that the actuator can move freely, the authors presented a greedy heuristic algorithm. This algorithm adds virtual nodes to the tree such that every tree edge is not longer than a pre-defined value. It iteratively selects as RPs the nodes with greatest utility, i.e., the nodes that will lead to greatest ratio of energy saving to length increase of the TSP tour of existing RPs. As new RPs are selected, already selected RPs whose utility becomes zero are removed. The selection process terminates when the maximum tour length is reached, or when all data sources are included.

Xing et al. [32] studied the trade off of energy consumption and communication delay by jointly optimizing the selection of RPs, actuator trajectory, and data transmission routes. They presented a heuristic algorithm based on Steiner Minimum Tree (SMT) [9]. A SMT of data sources is constructed with an arbitrary data source as root. It has minimum length and thus energy optimal for data communication. 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 actuator can travel within a given data collection deadline.

4.3 RP selection by clustering

Rao and Biswas [21] presented a generic data collection framework without location information. In this framework, a minimum k -hop dominating set is constructed. Nodes in the dominating set are called *navigation agents* (NA). Two adjacent NAs are at least $k + 1$ and at most $2k + 1$ hops away from each other. Each NA constructs a minimum hop tree rooted at itself and spanning up to a depth of $2k + 1$ hops. During tree construction, it identifies adjacent NAs and meanwhile constructs shortest paths to them. The nodes along such a shortest path are called *intermediate navigators* (IN). They are used to navigate the actuator to move between NAs.

NAs and INs constitute a connected overlay graph. An existing distributed TSP algorithm is adopted to find an actuator tour of NAs over the overlay graph. This algorithm enables each NA to know its next NA in the tour. Then the actuator starts to move from an arbitrary location to discover a local NA by listening to a hello message. Once the first NA is discovered, it moves toward the NA according to the received signal's Direction of Arrival (DOA). Afterwards, it travels along the actuator tour by following the DOA of signal of intermediate nodes. The immediate neighbors of a NA, called *designated gateways* (DG), are rendezvous points. Sources send data toward the actuator tour using NA-rooted trees. Data stops at the closest DG on its way. Along its TSP tour, the actuator retrieves data from encounters NAs and their DGs.

5 Conclusions

A more detailed elaboration on the surveyed algorithms in this paper can be found in [15]. These algorithms use different actuator mobility strategies and tackle the problem from different aspects. Almost all of them are centralized algorithms requiring full knowledge of the network. They do not scale well and have very limited applicability in practice, because WSN are usually deployed at random and full of dynamics (e.g., node failure and topological change). Future research is expected to focus on practical solutions that operate in decentralized or localized manner.

In current rendezvous-based data collection approaches, rendezvous points (RPs) stay unchanged once selected. Due to message relay overhead, uneven energy depletion will appear around RPs as the network evolves, offsetting the effectiveness of the algorithm for network lifetime elongation. Developing algorithms with dynamic RP selection will be an interesting future research direction.

Acknowledgments

This research was supported by the UK Royal Society Wolfson Research Merit Award, and NSERC Strategic Grants STPSC 356913-07 and STPGP 336406-08.

References

- [1] G. Anastasi, M. Conti, E. Gregori, C. Spagoni, and G. Valtente. Motes Sensor Networks in Dynamic Scenarios. *International Journal of Ubiquitous Computing and Intelligence*, 1(1), 2006.
- [2] G. Anastasi, M. Conti, E. Monaldi, and A. Passarella. “An Adaptive Data-transfer Protocol for Sensor Networks with Data Mules”. In *Proc. of IEEE WoWMoM*, pp. 1–8, 2007.
- [3] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. Wang. “Controlled sink mobility for prolonging wireless sensor networks lifetime”. *ACM Wireless Networks*, 14(6):831–858, 2008.
- [4] C. Buragohain, D. Agrawal, and S. Suri. “Power aware routing for sensor databases”. In *Proc. of IEEE INFOCOM*, vol. 3, pp. 1747–1757, 2005.
- [5] A. Chakrabarti, A. Sabharwal, and B. Aazhang. “Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks”. In *Proc. of IPSN*, vol. 2634 of *LNCS*, pp. 129–145, 2003.
- [6] Y. Gu, D. Bozdag, and E. Ekici. “Mobile Element Based Differentiated Message Delivery in Wireless Sensor networks”. In *Proc. of IEEE WoWMoM*, pp. 83–92, 2006.
- [7] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C.-G. Lee. “Partitioning Based Mobile Element Scheduling in Wireless Sensor Networks”. In *Proc. of IEEE SECON*, pp. 386–395, 2005.
- [8] S. Hashish and A. Karmouch. “Deployment-based Solution for Prolonging Lifetime in Sensor Networks with Multiple Mobile Sinks”. *Ad Hoc & Sensor Wireless Networks*, 2008. To appear.
- [9] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*. Elsevier, North-Holland, 1992.
- [10] D. Jea, A. Somasundara, and M. Srivastava. “Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks”. In *Proc. of IEEE DCOSS*, vol. 3560 of *LNCS*, pp. 244–257, 2005.
- [11] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin. “Intelligent Fluid Infrastructure for Embedded Networks”. In *Proc. of MobiSys*, pp. 111–124, 2004.
- [12] J. B. Kruskal. “On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem”. In *Proc. of the American Mathematical Society*, vol. 7, pp. 48–50, 1956.
- [13] E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys. *The traveling salesman problem*. John Wiley & Sons, 1985.
- [14] N. Li and J. Hou. “Topology control in heterogeneous wireless networks: problems and solutions”. In *Proc. of IEEE INFOCOM*, vol. 1, pp. 232–243, 2004.
- [15] X. Li, A. Nayak, and I. Stojmenovic. “Sink Mobility in Wireless Sensor Networks”. In X. Li, H. Liu, A. Nayak, and I. Stojmenovic, editors, *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*. Wiley, 2009. To appear.
- [16] X. Li and I. Stojmenovic. “Collecting Sensory Data by Map Exploration”, 2008. In preparation.
- [17] J. Lian, K. Naik, and G. Agnew. “Data Capacity Improvement of Wireless Sensor Networks Using Non-Uniform Sensor Distribution”. *International Journal of Distributed Sensor Networks*, 2(2):121–145, 2006.
- [18] J. Luo and J.-P. Hubaux. “Joint mobility and routing for lifetime elongation in wireless sensor networks”. In *Proc. of IEEE INFOCOM*, vol. 3, pp. 1735–1746, 2005.
- [19] S. Nesamony, M. K. Vairamuthu, and M. E. Orlowska. “On Optimal Route of a Calibrating Mobile Sink in a Wireless Sensor Network”. In *Proc. of INSS*, pp. 61–64, 2007.
- [20] S. Olariu and I. Stojmenovic. “Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting”. In *Proc. of IEEE INFOCOM*, pp. 1–12, 2006.
- [21] J. Rao and S. Biswas. “Joint Routing and Navigation Protocols for Data Harvesting in Sensor Networks”. In *Proc. of IEEE MASS*, pp. 143–152, 2008.
- [22] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: modeling and analysis of a three-tier architecture for sparse sensor networks. *Ad Hoc Networks*, 1(2–3):215–233, 2003.
- [23] S. Singh, M. Woo, and C. Raghavendra. “Power-aware routing in mobile ad hoc networks”. In *Proc. of ACM/IEEE MobiCom*, pp. 181–190, 1998.
- [24] I. Stojmenovic and X. Lin. “Power aware localized routing in wireless networks”. *IEEE Transactions on Parallel and Distributed Systems*, 12(11):1122–1133, 2001.
- [25] I. Stojmenovic and S. Olariu. “Data-Centric Protocols for Wireless Sensor Networks”. In I. Stojmenovic, editor, *Handbook of Sensor Networks*, pp. 417–456. Wiley, 2005.
- [26] R. Sugihara and R. K. Gupta. “Data mule scheduling in sensor networks: Scheduling under location and time constraints”. Technical Report CS2007-0911, CSE, University of California, San Diego, October 2007.
- [27] R. Sugihara and R. K. Gupta. “Improving the Data Delivery Latency in Sensor Networks with Controlled Mobility”. In *Proc. of IEEE DCOSS*, vol. 5067 of *LNCS*, pp. 386–399, 2008.
- [28] R. E. Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.
- [29] Z. Vincze, D. Vass, R. Vida, A. Vidacs, and A. Teles. “Adaptive sink mobility in event-driven densely deployed wireless sensor networks”. *Ad Hoc & Sensor Wireless Networks*, 3(2-3):255–284, 2007.
- [30] A. Wadaa, S. Olariu, L. Wilson, K. Jones, and M. El-toweissy. “Training a wireless sensor network”. *Mobile Networks and Applications*, 10(1-2):151–168, 2005.
- [31] X. Wu, G. Chen, and S. Das. “Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution”. *IEEE Transactions on Parallel and Distributed Systems*, 19(5):710–720, 2008.
- [32] G. Xing, T. Wang, W. Jia, and M. Li. “Rendezvous Design Algorithms for Wireless Sensor Networks with a Mobile Base Station”. In *Proc. of ACM MobiHoc*, pp. 231–239, 2008.
- [33] G. Xing, T. Wang, Z. Xie, and W. Jia. “Rendezvous Planning in Mobility-assisted Wireless Sensor Networks”. In *Proc. of IEEE RTSS*, pp. 311–320, 2007.