This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2009 proceedings.

Optimizing Energy-Latency Trade-off in Sensor Networks with Controlled Mobility

Ryo Sugihara Rajesh K. Gupta
CSE Department, University of California, San Diego, La Jolla, CA 92093, USA
Email: {ryo, rgupta}@ucsd.edu

***Abstract*—We consider the problem of planning path and speed of a "data mule" in a sensor network. This problem is encountered in various situations, such as modeling the motion of a data-collecting UAV (Unmanned Aerial Vehicle) in a field of sensors for structural health monitoring. Our specific context here is use of a data mule as an alternative or supplement to multihop forwarding in a sensor network. While a data mule can reduce the energy consumption at each sensor node, it increases the latency from the time the data is generated at a node to the time the base station receives it. In this paper, we introduce the "data mule scheduling" or DMS framework that enables data mule motion planning to minimize the data delivery latency. The DMS framework is general; it can express many previously proposed problem formulations and problem settings related to data mules. We design algorithms for DMS and extend to the more general case of combined data mule and multihop forwarding to enable a flexible trade-off between energy consumption and data delivery latency. Using DMS, we can calculate the optimal way for node-to-node forwarding and data mule motion plan. Our implementation and simulation results using ns2 show nearly monotonic decrease of data delivery latency when each node can use more energy, thus vastly increasing the flexibility in the energy-latency trade-off for sensor network communications.**

I. Introduction

Controlled mobility presents an attractive alternative to multihop forwarding for efficient data collection in a sensor field. In particular, we consider collecting data from stationary sensor nodes using a "data mule" via wireless communication. A data mule is a mobile node with radio and sufficient amount of storage to store the data from the sensors in the field. Data mules have been used in recent sensor network applications, e.g., a robot in underwater environmental monitoring [1] and a UAV (unmanned aerial vehicle) in structural health monitoring [2]. A data mule travels across the sensor field and collects data from each sensor node when the distance is short, and later deposits all the data to the base station. In this way, each sensor node can conserve energy, since it only needs to send the data over a short distance and has no need to forward other sensors' data all the way to the base station. Note that energy issue is critical for sensor nodes as opposed to the data mule that returns to the base station after the travel. However, one disadvantage of this approach is that it generally takes more time to collect data, which in turn incurs larger data delivery latency. Thus optimizing the data delivery latency is vital for the data mule approach to be useful in practice.

In this paper, we study the problem of optimizing the energy-latency trade-off when using a data mule. We design a problem framework for optimizing the data mule's movement, which we call the data mule scheduling (DMS) problem, and extend it to a general problem that combines data mule and multihop forwarding. Compared to previous studies, the DMS framework is comprehensive and general in the sense that it is capable of expressing many other formulations. It is also flexible enough to adapt to different problem settings.

In the DMS problem setting, we can control the movement of the data mule (path, speed) as well as its communication (i.e., from which node it collects data at certain time duration). There is some similarity to classical scheduling problems. For instance, the communication between the data mule and each node can be represented as a job that has both time and location constraints.

Then we consider the combined approach of data mule and multihop forwarding. In the pure data mule approach, the energy consumption at each node is minimum and the data delivery latency is relatively large. On the other hand, multihop forwarding requires greater energy due to increased data transfer at each node but the latency is expected to be much shorter. Our work combines these two approaches in such a way that the designers of sensor networks can balance the energy consumption and the data delivery latency according to application needs. We formulate the problem by extending the DMS problem and design an algorithm based on linear program formulation. Then we implement the combined approach on the ns2 network simulator [3] to experimentally evaluate the effectiveness of the formulation and algorithm.

The rest of this paper is organized as follows. In Section II we introduce related work. Section III gives an overview of the DMS problem. Section IV discusses the combined approach of data mule and multihop forwarding for the extended DMS problem. Section V shows the results of the simulation experiments on ns2 network simulator and Section VI concludes the paper.

II. Related Work

Use of mobile nodes for data collection have been explored in sensor networks. Somasundara et al. [4], [5] studied the problem of choosing the path of a data mule that traverses at a constant speed through a sensor field with sensors generating data at a given rate. Their formulation requires the data mule to visit the exact location of each sensor to collect data. In the Message Ferrying project, Zhao and Ammar [6] examined the problem of path and speed optimization of a data

978-1-4244-3513-5/09/$25.00 ©2009 IEEE

mule in a field of stationary nodes. The project has extended the work on controllably mobile nodes case [7], multiple data mules case [8], and arbitrarily mobile nodes case [9]. While these formulations are similar in spirit to ours, we also generalize the problem to include a precise mobility model with acceleration constraints and stronger guarantees on the optimality, as demonstrated in our previous study [10].

There are also studies on combined data mule and multi-hop forwarding approach. Ho and Fall [11] discussed such approach in the context of Delay Tolerant Networking (DTN) architecture. Burns et al. [12] experimentally showed that controlled mobility can improve performance of routing in a network of randomly mobile nodes.

Also relevant to this paper is work by Kansal et al. [13], who studied the case in which a data mule periodically travels across the sensor field along a fixed path. In their model, they can only change the speed of data mule. Their focus is on designing a robust communication infrastructure that works even in uncertain environments, whereas our work focuses on building a formal understanding of the problem and seeking for theoretical limits of performance.

Ma and Yang [14] designed a heuristic algorithm for path selection of a data mule. Similarly, Xing et al. [15] designed path selection algorithms when each node can forward data toward the base station along a routing tree constructed in advance. While these formulations are similar to ours in some ways, one of the limitations in these works is that they assume connected networks. Our problem framework can express not only their settings, but also more general settings including disconnected networks.

III. Problem Framework for Optimal Control of Data Mule

To control a data mule for data collection, one needs to determine the path and speed of the data mule and the schedule (i.e., when to collect data from a node). However, simultaneously optimizing them is an NP-hard problem, which is implied by the NP-hardness of the simplified path selection problem [10]. Consequently, in previous studies, the problem has been simplified in various ways by employing assumptions that restrict the capabilities of sensor nodes and data mule. Some examples are: data collection is only possible at the exact location of each node, and the data mule can move only at a constant speed. These assumptions are sometimes appropriate, but often make the formulation only applicable to a specific application and setting.

Our goal for designing the DMS problem is to provide a comprehensive and flexible problem framework in which we can fully exploit the networking and mobility capabilities. For this purpose, we first decompose the problem into following three subproblems (see Figure 1(b)(c)(d)):

1) Path selection: determines the trajectory of the data mule so that it travels within each sensor node's communication range at least once.
2) Speed control: determines how the data mule changes the speed along the path, so that it spends enough time within each node's communication range to collect all the data from it.
3) Job scheduling: determines the schedule of data collection from each node.

The last two subproblems are solved jointly as a scheduling problem with both location and time constraints. We call it the 1-D DMS problem and have presented algorithms in [16]. As for the path selection subproblem, we formulated it as an independent problem in [10], which we briefly describe later.

The DMS problem stated above is general and can be used to express several earlier problems in the area. For instance, the assumption of no remote communication (as in [4], [5]) is easily expressed by setting the communication range to zero in the path selection subproblem. The constant speed assumption (as in [14], [15]) and variable speed assumption (as in [6], [13]) are handled in the speed control subproblem.

In the next section, we extend the DMS problem to broaden the coverage for more general cases. Specifically, we consider the hybrid case that combines data mule with multihop forwarding. We realize this by adding the "forwarding" subproblem in front of the path selection subproblem, as shown in Figure 1(a).

IV. Combining DMS with Multihop Forwarding

Using the DMS problem, we are able to determine data mule's path, speed and schedule for sensor data collection so that the travel time is minimized. We now consider a combined approach of data mule and multihop forwarding. In our framework, we realize this by defining a new "forwarding" problem that is placed in front of the DMS problem as shown in Figure 1. The forwarding problem is to determine how much data each node forwards to other nodes and to the data mule while satisfying a predetermined energy consumption constraint.

First we consider the path selection problem. Then we discuss the forwarding problem and present a linear program formulation.

A. Overview of path selection problem

For the nodes to send their data to the data mule, a path needs to intersect with their communication ranges. The objective is to find a path such that the shortest travel time of the data mule in the 1-D DMS problem induced by that path is minimized. However, finding a "smooth" path is computationally expensive and maneuvering the data mule along such a smooth path is often difficult as well. From these reasons, we have designed and analyzed a simplified path selection problem in [10].

We consider a complete graph having vertices at sensor nodes' locations and assume the data mule moves between vertices along a straight line. Each edge is associated with a cost and a set of labels, where the latter represents the set of nodes whose communication ranges intersect with this edge. In this way, while traveling along an edge, the data mule can collect data from the nodes in the labels associated with it. The objective is to find a "label-covering tour" that minimizes

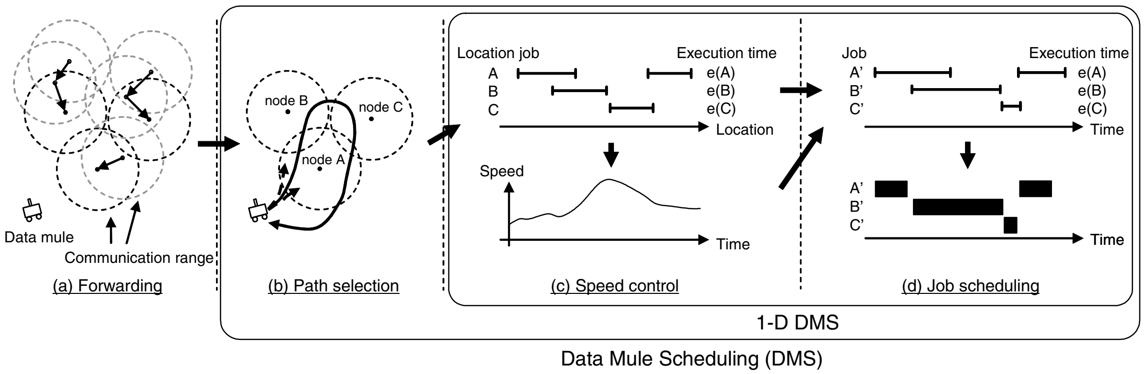

Fig. 1. Subproblems of the DMS problem: Forwarding problem (discussed in Section IV) formulates the combined approach of data mule and multihop forwarding.

the total cost of the edges in the tour, where "label-covering" means that, for any label, there exists at least one edge in the tour that contains the label. We use Euclidean distance as the cost metric, since we have observed in the experiments that it has a strong positive correlation with the shortest travel time in the induced 1-D DMS problem.

The simplified problem is still NP-hard and we have designed an approximation algorithm. The algorithm first finds a TSP tour by using any TSP solver as an external module. Then, using dynamic programming, it finds a short label-covering tour that can be constructed by shortcutting the TSP tour, which is also a label-covering tour by itself. The algorithm runs in $\mathcal{C}_{TSP} + O(n^3)$ time, where $\mathcal{C}_{TSP}$ is the computation time of the TSP solver. An approximate label-covering tour T_{APP} found by this algorithm satisfies $|T_{APP}| \le \alpha(|T_{OPT}| + 2nr)$, where α is the approximation ratio of the TSP solver, T_{OPT} is the shortest label-covering tour, n is the number of nodes, and r is the radius of communication range. In [10], simulation experiments have demonstrated that our formulation and algorithm effectively exploit broader communication range and yield shorter travel time than previous studies such as [6], [14], [15].

B. Forwarding problem

The objective of the forwarding problem is to find a forwarding plan such that the induced DMS problem has the shortest total travel time. Different from the "pure" data mule approach, in which each node sends its data only to the data mule, it can now forward its data to other neighboring nodes as well. More importantly, if a node decides to forward all data to other nodes, the data mule does not need to collect data directly from this node. Then the data mule can possibly take a shorter path to reduce the travel time.

We present a centralized algorithm based on linear program formulation. Since finding the optimal forwarding plan that minimizes the travel time in the induced DMS problem is at least as hard as the DMS problem, we make it an independent problem by changing the objective function.

As the objective, we choose to minimize the average distance of nodes from the base station weighted by the amount of data at each node after forwarding. There are three reasons why this is a reasonable choice. First, this function is likely to shorten the path of the data mule by forcing the nodes at the edge of network to primarily use forwarding. Secondly, this function allows a smooth transition between the data mule approach and multihop forwarding. As the energy consumption limit grows, more data is forwarded closer to the base station. In a connected network, all the data is eventually forwarded to the base station without using a data mule, which is equivalent to "pure" multihop forwarding. Finally, since the function is linear, we can formulate the problem as a linear program as described below.

We assume the location of sensor nodes and the connectivity between them are known. We also assume the following parameters are given:

- λ_i: Data generation rate of node i
- E_{limit}: Energy consumption limit at each node per unit time
- E_r, E_s: Energy consumption for receiving and sending unit data
- R: Bandwidth, i.e., maximum data rate that each node can communicate with other nodes and the data mule

Then we have the following linear program:

Variables

- x_{ij}: Amount of data sent from node i to j per unit time

Objective Minimize $\sum_i d_i \lambda'_i$, where d_i is the distance between node i and the base station, and λ'_i is the data rate that node i sends directly to the data mule. We have $\lambda'_i = \sum_j x_{ji} + \lambda_i - \sum_j x_{ij}$, which is the difference of incoming data rate and outgoing data rate.

Constraints

- $x_{ii} = 0$.
- (Connectivity) For $i \ne j$, $x_{ij} \ge 0$ if node j is in the communication range of node i. Otherwise $x_{ij} = 0$.
- (Flow conservation) $\lambda'_i \ge 0$.
- (Energy consumption) For each node i,

$$E_r \sum_j x_{ji} + E_s \left(\sum_j x_{ji} + \lambda_i \right) \le E_{limit},$$

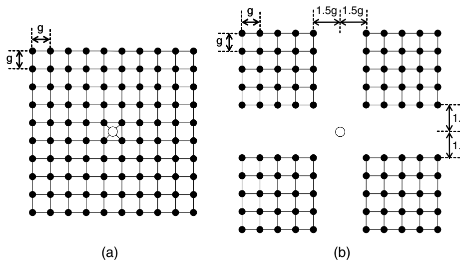

Fig. 2. Network topology: (a) Connected network; (b) Disconnected network. White circle is the base station. Line between two circles represents that they are within the communication range. Grid size g is set to $0.8r$, where r is the radius of communication range, and a uniformly random disturbance of $[-0.025r, 0.025r]$ is added to the position of each node.

where the first term in the left hand side is the amount of energy consumed by receiving data and the second term is that for sending data. About the latter, node i sends $\sum_j x_{ij}$ to other nodes and λ'_i to the data mule per unit time, when averaged over time, and the sum of these two equals $\sum_j x_{ji} + \lambda_i$.

- (Bandwidth) Per unit time, the amount of incoming data is $\sum_j x_{ji}$ and outgoing data is $\sum_j x_{ij} + \lambda'_i$. After some manipulations, we obtain

$$2\sum_j x_{ji} + \lambda_i \leq R.$$

The formulation above is also capable of expressing the case in which each node communicates along the preconstructed routing tree as in [15]. This is possible by replacing the connectivity constraint with the following one:

- (Routing tree) For $i \neq j$, $x_{ij} \geq 0$, if node j is node i's parent in the routing tree. Otherwise $x_{ij} = 0$.

V. Simulation Experiments

We experimentally evaluate the combined approach of data mule and multihop forwarding in the periodic data generation case, specifically on the effectiveness of formulation and algorithms in optimizing the energy-latency trade-off.

A. Methods

We have implemented the algorithm for the forwarding problem and the algorithms for the DMS problem in MATLAB. The MATLAB program generates a Tcl script for ns2 [3], which simulates the movement of the data mule and the communication among the data mule and the nodes.

To assess performance, we measure the delivery latency for each data packet from the time it is generated to the time the base station receives it either from neighboring nodes or via the data mule. For each test case, the simulation on ns2 is repeated multiple periods until it reaches stability. We consider it stable when the average delivery latency of the data received in the current period is within $\pm 1\%$ of that of the previous period. If it is stable, we use the data for the next period as the final results.

Figure 2 shows the network topologies we use for the experiments. Both of them have 100 sensor nodes, one base station and one data mule, but one is a connected network and the other is a disconnected network. The disconnected network consists of four connected networks of 25 nodes and the base station is not directly reachable from any nodes.

For the data mule's movement, we use the variable speed model. The range of speed is $0 \leq v \leq 10m/s$, which roughly simulates the movement of a UAV used in [2].

For ns2 simulator, we use FreeSpace propagation model with 100m communication range. We use 802.11 MAC (with RTS/CTS) with 2 Mbps raw bandwidth, which is the default value for ns2. Packet size is 400 Bytes.

Other parameters are set as follows. Energy consumption for sending/receiving unit data is assumed to be equal, i.e., $E_r = E_s$. The rate of data generation at each node λ_i is 100 Byte/sec. Let E denote the energy consumption at each node for "pure data mule" case without any node-to-node forwarding. Then E is expressed as $\lambda_i E_s$, and this is the minimum possible value of E_{limit}. We measured the latency for $E_{limit} = E, ..., 50E$. Effective bandwidth R is set to 400 Kbps, considering the overhead of RTS/CTS and packet header.

B. Results

In ns2 simulation, the delivery latency reached stability in all tested cases. Average number of periods until reaching stability was 5.2 in the connected network, and 4.0 in the disconnected network.

Figure 3 shows the simulation results for the connected network and disconnected network. For both networks, $E_{limit} = E$ corresponds to the "pure data mule" case. The average latency in this case was 432.81 secs for the connected network and 513.07 secs for the disconnected network. These are very close to the data mule's travel time (440.73 secs and 511.68 secs, respectively). For the connected network, it became "pure multihop forwarding" when $E_{limit} = 49E$, where all the data are sent to the base station solely by multihop forwarding and the data mule is not used. The average latency in this case was 4.17 secs.

C. Discussions

For both of the connected and the disconnected networks, the simulation results showed the decrease of data delivery latency as the energy consumption limit increases. The decrease was almost monotonic, demonstrating fine-grained control of the trade-off between energy and latency.

We can also observe that the travel time of the data mule is nearly equal to the average latency for the data delivered by the data mule. It demonstrates that minimizing the travel time for the purpose of minimizing the data delivery latency is a valid approach. In addition, this implies that we can estimate the average delay by solving the DMS problem.

Figure 4 shows the histograms of data delivery latency for different energy consumption limits. As these figures show, regardless of the different network topology and the different

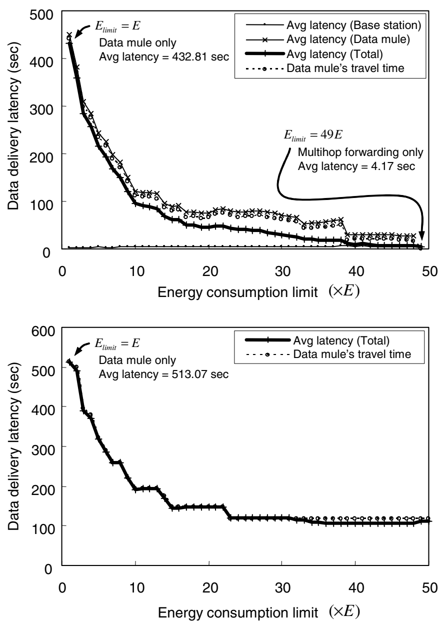

Fig. 3. Data delivery latency for varying energy consumption limit: (top) connected network, (bottom) disconnected network.

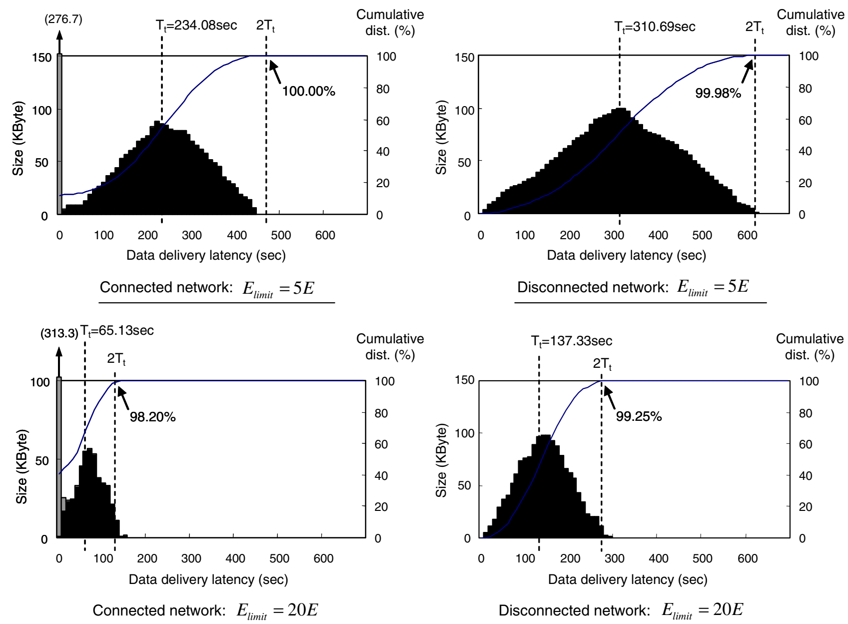

Fig. 4. Histogram of data delivery latency: Data mule's travel time (T_t) is also shown. For the connected network, gray bars are for the data delivered to the base station from neighboring nodes, and black bars are for the data delivered via the data mule.

total travel time, more than 98% of the data has delivery latency within double of the travel time. This means we can estimate the maximum delivery latency as well as the average.

VI. Conclusion

Controlled mobility, as represented by the motion of a data mule, provides an alternative approach to multihop forwarding for collecting data from sensor networks. While it allows a significant reduction in energy consumption, increased data delivery latency is a big issue. In this paper, we have presented the data mule scheduling (DMS) problem as a problem framework for optimally controlling a data mule and have extended it to enable flexible energy-latency trade-off. We have presented a framework to capture and analyze communication strategies that use combinations of data mule and multihop forwarding. To validate our results, we have implemented our algorithms and simulated them on ns2 network simulator. The results showed nearly monotonic decrease of the data delivery latency for larger energy consumption limit, demonstrating the effectiveness of the formulation and the algorithms in optimizing the energy-latency trade-off.

Acknowledgment

This work was supported in part by UCSD/LANL Engineering Institute and by NSF under grant numbers CCF-0702792 and SRS-0820034.

References

[1] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in *SenSys*, 2005, pp. 154–165.

[2] M. Todd, D. Mascarenas, E. Flynn, T. Rosing, B. Lee, D. Musiani, S. Dasgupta, S. Kpotufe, D. Hsu, R. Gupta, G. Park, T. Overly, M. Nothnagel, and C. Farrar, "A different approach to sensor networking for SHM: Remote powering and interrogation with unmanned aerial vehicles," in *Proceedings of the 6th International workshop on Structural Health Monitoring*, 2007.

[3] "ns2 network simulator," http://www.isi.edu/nsnam/ns/.

[4] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in *RTSS*, 2004, pp. 296–305.

[5] ——, "Mobile element scheduling with dynamic deadlines," *IEEE Trans. Mobile Computing*, vol. 6, no. 4, pp. 395–410, 2007.

[6] W. Zhao and M. Ammar, "Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks," in *FTDCS*, 2003, pp. 308–314.

[7] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *MobiHoc*, 2004, pp. 187–198.

[8] ——, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *INFOCOM*, 2005, pp. 1407–1418.

[9] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *MobiHoc*, 2006, pp. 37–48.

[10] R. Sugihara and R. K. Gupta, "Improving the data delivery latency in sensor networks with controlled mobility," in *DCOSS*, 2008.

[11] M. Ho and K. Fall, "Poster: Delay tolerant networking for sensor networks," in *SECON*, 2004.

[12] B. Burns, O. Brock, and B. N. Levine, "MV routing and capacity building in disruption tolerant networks," in *INFOCOM*, 2005, pp. 398–408.

[13] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networks," in *MobiSys*, 2004, pp. 111–124.

[14] M. Ma and Y. Yang, "SenCar: An energy efficient data gathering mechanism for large-scale multihop sensor networks," *IEEE Trans. Parallel and Distributed System*, vol. 18, no. 10, pp. 1476–1488, 2007.

[15] G. Xing, T. Wang, Z. Xie, and W. Jia, "Rendezvous planning in mobility-assisted wireless sensor networks," in *RTSS*, 2007, pp. 311–320.

[16] R. Sugihara and R. K. Gupta, "Scheduling under location and time constraints for data collection in sensor networks," in *RTSS (work-in-progress session)*, 2007.