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

Energy-efficient Itinerary Planning for Mobile Agents in Wireless Sensor Networks

†Min Chen, †Victor Leung, ‡Shiwen Mao, *Taekyoung Kwon and §Ming Li
†Dept. of Electrical & Computer Engineering, Univ. of British Columbia, V6T 1Z4, Canada (minchen,vleung@ece.ubc.ca)
‡Dept. of Electrical & Computer Engineering, Auburn Univ., Auburn, AL 36849-5201, USA (smao@ieee.org)
* School of Computer Science & Engineering, Seoul National University, Seoul, 151-744, Korea (tkkwon@snu.ac.kr)
§Dept. of Computer Science, California State Univ., Fresno, CA 93740, USA (mingli@csufresno.edu)

***Abstract*—Compared to conventional wireless sensor networks (WSNs) that are operated based on the client-server computing model, mobile agent (MA) systems provide new capabilities for energy-efficient data dissemination by flexibly planning its itinerary for facilitating agent based data collection and aggregation. It has been known that finding the optimal itinerary is NP-hard and is still an open area of research. In this paper, we consider the impact of both data aggregation and energy-efficiency in sensor networks itinerary selection, We propose an *itinerary energy minimum for first-source-selection* (IEMF) algorithm, as well as the *itinerary energy minimum algorithm* (IEMA), the iterative version of IEMF. Our simulation experiments show that IEMF provides higher energy efficiency and lower delay compared to existing solutions, and IEMA outperforms IEMF with some moderate increase in computation complexity.**

I. Introduction

The application-specific nature of tasking a wireless sensor network (WSN) requires that sensor nodes have various capabilities for multiple applications. It would be impractical to store in the local memory of embedded sensors *all* the programs needed to run every possible application, due to the tight memory constraints. A *mobile agent* (MA) is a special kind of software that migrates among network nodes to carry out task(s) autonomously and intelligently in response to changing conditions in the network environment, in order to achieve the objectives of the agent dispatcher. The use of MAs to dynamically deploy new applications in WSNs, has been proven to be an effective method to address this challenge.

Recently there has been a growing interest on the design, development, and deployment of MA systems for high-level inference and surveillance in WSNs [1]–[8]. In [1], the agent design in WSNs is decomposed into four components, i.e., architecture, itinerary planning, middleware system design and agent cooperation. Among the four components, itinerary planning determines the order of source nodes to be visited during agent migration, which has a significant impact on energy performance of the MA system. It has been shown that finding an optimal itinerary is NP-hard. Therefore, heuristic algorithms are generally used to compute competitive itineraries with a sub-optimal performance.

In [2], two simple heuristics are proposed: (i) a *local closest first* (LCF) scheme that searches for the next node with the shortest distance to the current node, and (ii) a *global closest first* (GCF) scheme that searches for the next node closest to the dispatcher. These two schemes only consider the spatial distances between sensor nodes and thus, may not be energy efficient in many cases. A genetic algorithm (GA) [3] is proposed to exploit the global information of sensor detection signal levels and link power consumption. In GA, every node reports its status to the sink node, which may incur considerable control overhead. It is neither scalable to network size nor a lightweight solution that is suitable for sensor nodes constrained in energy supply. The original LCF, GCF [2] and GA schemes [3] are all based on the following two assumptions: (i) a cluster-based network architecture, where all nodes (e.g., sink and source nodes) can communicate with each other in one hop; (ii) high redundancy among the sensory data, which can be fused into a single data packet with a fixed size. This implies that a *perfect aggregation model* is used. These assumptions limit the scope of the existing schemes.

In this paper, we focus on designing lightweight, energy efficient itinerary planning algorithms without making the above assumptions. We first propose an *itinerary energy minimum selection for first-source-selection* (IEMF) algorithm, which extends LCF by choosing the first source node to visit based on estimated communication cost. In IEMF, the impact of both data aggregation and energy efficiency are taken into account to obtain an energy-efficient itinerary. The scheme is quite general, in the sense that it adopts a universal aggregation model, which facilitates the support for a wide range of applications. In addition, IEMF does not rely on any specific network architecture and is suitable for multi-hop WSNs. We also observe that IEMF achieves energy efficient itineraries without incurring additional control overhead, as compared with existing lightweight approaches such as LCF and GCF.

Furthermore, we propose the *itinerary energy minimum algorithm* (IEMA), which is an iterative version of IEMF. During each iteration, IEMA selects an optimal source node as the next source to visit among the remaining set of source nodes. We show that with more iterations, the suboptimal itinerary can be progressively improved, while the largest reduction in average delay and energy consumption are achieves after the first few iterations. We can thus trade off between energy efficiency and computational complexity based on specific application requirements.

The remainder of the paper is organized as follows. The problem is stated in Section II. We present IEMF and IEMA in Section III. Our simulation studies are reported in Section IV. Section V concludes the paper.

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

II. Problem Statement

A. Aggregation Model

Consider an MA dispatched by the sink node to collect data from n source nodes. Let S_{proc} be the size of the MA processing code, S_{head} the size of agent packet header, and S^0_{ma} the agent size when it is first dispatched by the sink node. Then we have $S^0_{ma} = S_{proc} + S_{head}$. Let $r \in [0, 1)$ be the *reduction ratio* in sensory data by agent assisted local processing and S_{data} be the size of raw data at a source node. The reduced data payload collected by the agent at each source, denoted as S_{rd}, is $S_{rd} = (1 - r) \cdot S_{data}$. Let S^k_{ma} be the agent size when it leaves the kth source ($1 \leq k \leq n$). Since there is no data aggregation at the first source, we have $S^1_{ma} = S^0_{ma} + S_{rd}$.

Since the agent visits the second source node, it begins to perform aggregation to reduce the redundancy between the data collected in the source and the data it carries. Let $\rho \in [0, 1]$ denote the *aggregation ratio*, a measure of the compression performance. The MA size after it leaves the second source node is $S^2_{ma} = S^0_{ma} + S_{rd} + (1-\rho)S_{rd}$, and so forth. For the sake of simplicity, we assume that r, ρ and S_{data} are identical at each source.[1] After visiting the kth source node, we have

$$\begin{aligned} S^k_{ma} &= S^{k-1}_{ma} + (1-\rho)S_{rd} \\ &= S^0_{ma} + [1 + (k-1)(1-\rho)]S_{rd}. \end{aligned} \quad (1)$$

After visiting all the n source nodes, the MA has a size S^n_{ma} in the range $\left[S^0_{ma} + S_{rd}, S^0_{ma} + nS_{rd}\right]$. The lower bound $S^0_{ma} + S_{rd}$ corresponds to a perfect aggregation model where multiple packets are compressed in to a single one, while the upper bound $S^0_{ma} + nS_{rd}$ corresponds to the case of no aggregation performed at the MA.

B. A Generic Itinerary Planning Algorithm

We state our assumption and define a generic itinerary planning algorithm in this section. Specifically, we assume that the set of source nodes to visit is predetermined. In addition, the location information of the source nodes is also available at the sink node [2], [3]. Under these assumptions, we actually consider static itinerary planning [1], where an energy efficient itinerary is chosen based on the location information of the source nodes. Note that these are the general assumptions made in all of itinerary planning algorithms discussed in this paper.

Let n represents the number of source nodes and $V(n) = \{Src_i | Src_i = \text{the } i\text{th source}, i \in [1, 2, .., n]\}$ denote the set of source nodes to be visited by an MA. Also let s and t be the starting and ending point of the agent, respectively. we define a generic static itinerary planning algorithm as a function $F[s, V(n), t]$. Since the ending point is always the sink node, we abbreviate the function as $F[s, V(n)]$. Although usually s is set to the sink node, it can also be set to one of the source nodes when defining an iterative algorithm (see Section III-D). Specifically, we define a generic static itinerary planning algorithm with the following components:

- *Next-source-selection Algorithm*: denoted as $f[s, V(n)]$. Given a source list $V(n)$ and a starting point s, it selects the best source node to visit next, denoted as Src_{next}, among the n candidates source nodes.
- *Static Itinerary Planning Algorithm*: denoted as $F[s, V(n)]$. Given $V(n)$, s, and a next-source-selection function $f[s, V(n)]$, it computes an itinerary $V_{ip}(n) = \{Src^1_{ip}, Src^2_{ip}, \cdots, Src^n_{ip}$, based on a specific itinerary planning criterion.

Consider LCF for example [2], where $lcf[s, V(n)]$ is the LCF next-source-selection function and the output of $lcf[s, V(n)]$ is the best next source node Src^{lcf}_{next}. Furthermore, $LCF[s = t, V(n)]$ is the LCF function which returns the entire itinerary, denoted as $V^{lcf}_{ip}(n)$.

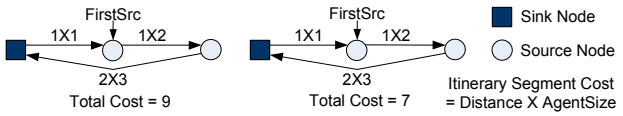


Fig. 1. Choosing different first source nodes results in different total costs.

III. The Energy Efficient Itinerary Planning Algorithms

A. Motivation

The simple example shown in Fig. 1 illustrates the motivation for IEMF. There are three nodes in the chain of the WSN, the sink and two source nodes. Based on LCF, the node in the middle will be visited first. Assuming the distance is one between two adjacent nodes. Let the original agent size be 1, and assume the agent size is increased by 1 after visiting each of the sources. We can calculated the *itinerary segment cost* by multiplying the *itinerary segment distance* with the *current agent size*. The resulting total cost of LCF is 9. If the agent visits the other source node first, the total cost is decreased to 7. Thus LCF performance can be improved by carefully choosing the first source node in the itinerary.

B. Estimated Communication Cost of a Candidate Itinerary

We first show how to estimate the communication cost of a given itinerary $\{t \rightarrow V_{ip}(n) \rightarrow t\} = \{t, \{Src^1_{ip}, Src^2_{ip}, \cdots, Src^n_{ip}\}, t\}$, which means an agent starts from sink t and returns back to t after migration. Generally, the communication energy consumption for receiving a data packet consists of the receiving energy, the control energy, and the transmitting energy. Let e_{ctrl} be the energy spent on control messages exchanged for a successful data transmission. Let m_{tx} and m_{rx} be the energy consumption for receiving and transmitting a data bit, respectively. Without loss of generality, we assume that m_{tx}, m_{rx} and e_{ctrl} are identical at each node without power control. Let S_{rx} and S_{tx} be the size of a received packet and that of a transmitted packet. The communication energy consumption at a specific node can be denoted by

$$e(S_{rx}, S_{tx}) = m_{rx} \cdot S_{rx} + m_{tx} \cdot S_{tx} + e_{ctrl}. \quad (2)$$

[1] The case of heterogeneous r_i, ρ_i and $S_{data,i}$ at each source node i can be easily handled in a similar fashion.

Multiple hops may exist between two adjacent source nodes, Src_{ip}^{k-1} and Src_{ip}^{k}. Let $d(k-1,k)$ denote the distance between the two source nodes. In a dense WSN, we can estimate the hop count between Src_{ip}^{k-1} and Src_{ip}^{k} as $H_{k-1}^{k} = \lceil \frac{d(k-1,k)}{R} \rceil$, where R represents the maximum transmission range. When the agent traverses intermediate sensor nodes, its size remains the same; its size may be increased after visiting each of the source nodes. Let $E_{k-1}^{k}(S_{ma}^{k-1})$ be the communication energy consumed when the MA roams from Src_{ip}^{k-1} to Src_{ip}^{k} with size S_{ma}^{k-1}. We estimate the communication energy cost as:

$$\begin{aligned} E_{k-1}^{k}(S_{ma}^{k-1}) &= m_p \cdot S_{data} + e(0, S_{ma}^{k-1}) \\ &+ H_{k-1}^{k} \cdot e(S_{ma}^{k-1}, S_{ma}^{k-1}) + e(S_{ma}^{k-1}, 0), \quad (3) \end{aligned}$$

where $m_p \cdot S_{ma}^{k-1}$ is the data processing energy at Src_{ip}^{k-1}; $e(0, S_{ma}^{k-1})$ is the energy for Src_{ip}^{k-1} to transmit the agent; $H_{k-1}^{k} \cdot e(S_{ma}^{k-1}, S_{ma}^{k-1})$ is the energy consumption of intermediate sensor nodes between Src_{ip}^{k-1} and Src_{ip}^{k}; and $e(S_{ma}^{k-1}, 0)$ is the energy consumption for Src_{ip}^{k} to receive the agent.

We divide the whole itinerary into three phases.

1) *Code-conveying phase*: the phase when the processing code is conveyed to the target region, during which the MA migrates from the sink to the first source node Src_{ip}^{1}. The communication energy consumption in this phase is denoted by E_{conv}, i.e., $E_{conv} = H(t, Src_{ip}^{1}) \cdot e(S_{ma}^{0}, S_{ma}^{0})$.
2) *Roaming phase*: starting from the time when MA leaves the first source node Src_{ip}^{1} to the time when it visits the last source node Src_{ip}^{n}. The communication energy consumption in this phase is denoted by E_{roam}, as $E_{roam} = \sum_{k=2}^{n} E_{Src_{ip}^{k-1} \rightarrow Src_{ip}^{k}}(S_{ma}^{k-1})$.
3) *Returning phase*: starting from the time when MA finishes visiting all the source nodes to the time when it returns to the sink. The communication energy consumption in this phase is denoted by E_{back}, i.e., $E_{back} = m_p \cdot S_{data} + e(0, S_{ma}^{n}) + H(Src_{ip}^{n}, t) \cdot e(S_{ma}^{n}, S_{ma}^{n})$, where $m_p \cdot S_{data}$ is the data processing energy at the last source node Src_{ip}^{n}; $e(0, S_{ma}^{n})$ is the energy for Src_{ip}^{n} to transmit the agent; and $H(Src_{ip}^{n}, t) \cdot e(S_{ma}^{n}, S_{ma}^{n})$ is the energy consumption of intermediate sensor nodes between the last source node and the sink node.

Finally, the estimated communication energy of a specific itinerary $\{t \rightarrow V_{ip}(n) \rightarrow t\}$ is $E_{itinerary} = E_{conv} + E_{roam} + E_{back}$. We further define an itinerary $I[s, V_{ip}(n)]$ as a function of starting point s and the sorted source sequence $V_{ip}(n)$. It starts from s, travels to each source node in $V_{ip}(n)$ one after the other, and ends at the sink node. The communication cost of the itinerary is denoted by $E_{I[s,V_{ip}(n)]}$. The ending point is always the sink node, while s can be set to one of the source nodes for computing the cost of a segment of the itinerary.

C. The IEMF Algorithm

Among the n source nodes in $V(n)$, different algorithms select various source node as Src_{ip}^{1}. For example, LCF and GCF select the one which is the closest to the sink as Src_{ip}^{1} [2], while MADD selects the farthest source node as Src_{ip}^{1} [4]. While these prior work can be categorized as pure distance-based selection, IEMF considers both aggregation ratio and energy efficiency to select the source node as Src_{ip}^{1} which yields the minimum itinerary cost.

Specifically, IEMF algorithm first select an arbitrary source node Src_i as the tentative Src_{ip}^{1}. The remaining source set is denoted as $V^i(n-1)$. Next, Src_i is set as the start point and the LCF criterion is used to determine the itinerary for the $n-1$ source nodes in $V^i(n-1)$. Executing function $LCF[Src_i, V^i(n-1)]$, we can get the source visiting sequence $V_{ip}^{lcf}(n-1)$. Then, the entire itinerary sequence starting from the sink can be obtained: $\{t \rightarrow Src_i \rightarrow V_{ip}^{lcf}(n-1) \rightarrow t\}$. The estimated cost of this itinerary is $E_{itinerary}^{i} = E_{[t,V_{ip}^{i}(n)]}$, where $V_{ip}^{i}(n) = \{Src_i, V^i(n-1)\}$.

Choosing each source in $V(n)$ as tentative Src_{ip}^{1} in a round robin fashion, we can get n different candidate itineraries and their corresponding itinerary costs. Among the n candidates, IEMF selects the one that has the minimum itinerary cost.

D. The IEMA Algorithm

IEMF selects the first source Src_{ip}^{1} as the one whose corresponding itinerary yields the smallest communication cost among the n candidate itineraries. Once Src_{ip}^{1} is determined, the corresponding itinerary is actually determined using the LCF criterion [2]. In this section, we propose an iterative version of IEMF, termed the *itinerary energy minimum algorithm* (IEMA). In addition to optimize Src_{ip}^{1}, IEMA also optimizes the remaining source nodes along the entire itinerary.

Let κ denote the number of iterations in IEMA. Since each iteration optimizes one source-selection, we have $\kappa \in [1, n]$. We denote IEMA(κ) as IEMA with κ iterations. Specifically, LCF and IEMF are the two special cases of IEMA: LCF is the 0 iterative version of IEMA, i.e., IEMA(0), and IEMF is the 1 iteration version of IEMA, i.e., IEMA(1).

Given a specific κ, IEMA(κ) only optimizes the *first* κ source nodes using the basic IEMF method. The remaining $n-\kappa$ source node will be simply sorted through LCF method. Clearly κ provides a convenient trade-off between energy savings and computational complexity.

IV. Performance Evaluation

A. Simulation Setting

We implement the proposed algorithms as well as the three existing algorithms (LCF, GCF and GA) using OPNET Modeler, and perform extensive simulations. We choose a network where nodes are uniformly deployed within a 1000m $\times$ 500m field. To verify the scaling property of DCF, we select a large-scale network size with 800 nodes. We let the sink node be located at the right side of the field and multiple source nodes be randomly distributed in the network.

The sensor application module consists of a constant-bit-rate source, which generates a sensed data report every 1 s (1024 bits each). As in [10], we use IEEE 802.11 DCF as the underlying MAC, and the radio transmission range (R) is set to 60 m. The data rate of the wireless channel is 1

TABLE I
SIMULATION PARAMETERS

Raw Data Reduction Ratio (r)	0.5
Aggregation Ratio (ρ)	0.5
MA Accessing Delay (τ)	10 ms
Data Processing Rate (V_p)	50 Mbps
Size of Sensed (Raw) Data (S_{data})	Default: 2048 bits
Size of Processing Code (S_{proc})	1024 bits
The Number of Source Node (N)	Default: 9

Mb/s. All messages are 64 bits in length. For consistency, we use the same energy consumption model as in [10], [11]. The initial energy of each node is 5 Joules. The transmit, receive and idle power consumptions are 0.66 W, 0.395 W, and 0.035 W, respectively. We count all types of energy consumptions in the simulations, including transmission, receiving, idling, overhearing, collisions and other unsuccessful transmissions, MAC layer headers, retransmissions, and RTS/CTS/ACKs.

We consider the following two performance metrics: (i) *Average Report Delay*: average delay from the time when MA is dispatched by the sink to the time when the agent returns to the sink. (ii) *Average Communication Energy*: the total communication energy consumption, including transmitting, receiving, retransmissions, overhearing and collision, over the total number of distinct reports received at the sink.

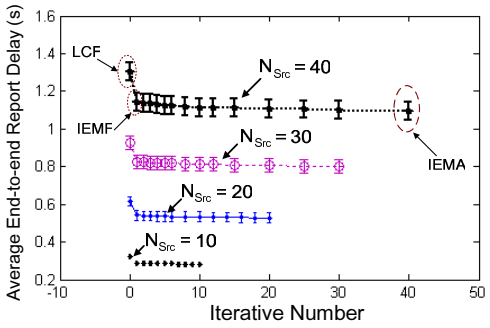
In all the figures presented in this section, each data point is the average of 45 simulation runs with different random seeds. Under each random seed, all the source nodes are randomly relocated. In all the figures, we plot the 95% confidence interval for each data point.

B. Simulation Results

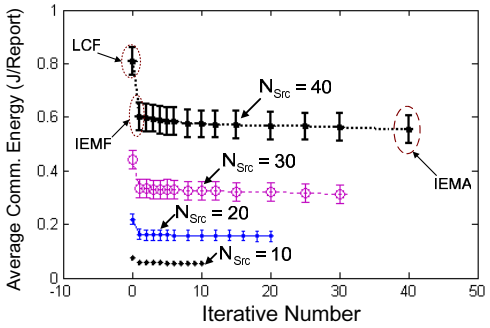
We first examine the impact of the number of iterations κ on the IEMA(κ) performance. We set the number of source nodes n to 10, 20, 30 and 40, and obtain a set of results for each case. For a given n value, we randomly choose the set of source nodes in each simulation with a different random seed.

In Fig. 2, we plot the average end-to-end delay for different iteration number κ, while in Fig. 2(b), we plot the average communication cost for different κ values. As expected, the number of source nodes n has a big impact on the delay and energy performance; both delay and energy consumption are much larger for a larger n value. This is because a large n means more source nodes to visit. The MA size will be larger and more transmissions will be made. Actually n is a good indicator of the MA related traffic load. From the descriptions of the algorithms, IEMA(0) is actually equivalent to LCF, and IEMA(1) is equivalent to IEMF, as indicated in the figures.

We also find that all the curves are monotonically decreasing with κ. However the largest reduction in both delay and energy consumption are achieved in the first few iterations, while the reductions in delay and energy become smaller and smaller as κ gets large. Therefore, the simple algorithm IEMF can achieve pretty good performance in many cases. Generally, we can determine a suitable κ for given application QoS requirements; there is no need to have n iterations in many cases. In contrast, we find the GA algorithm [3] achieves visible improvement only after about 100 iterations. Such



(a) Average end-to-end delay



(b) Average communication energy

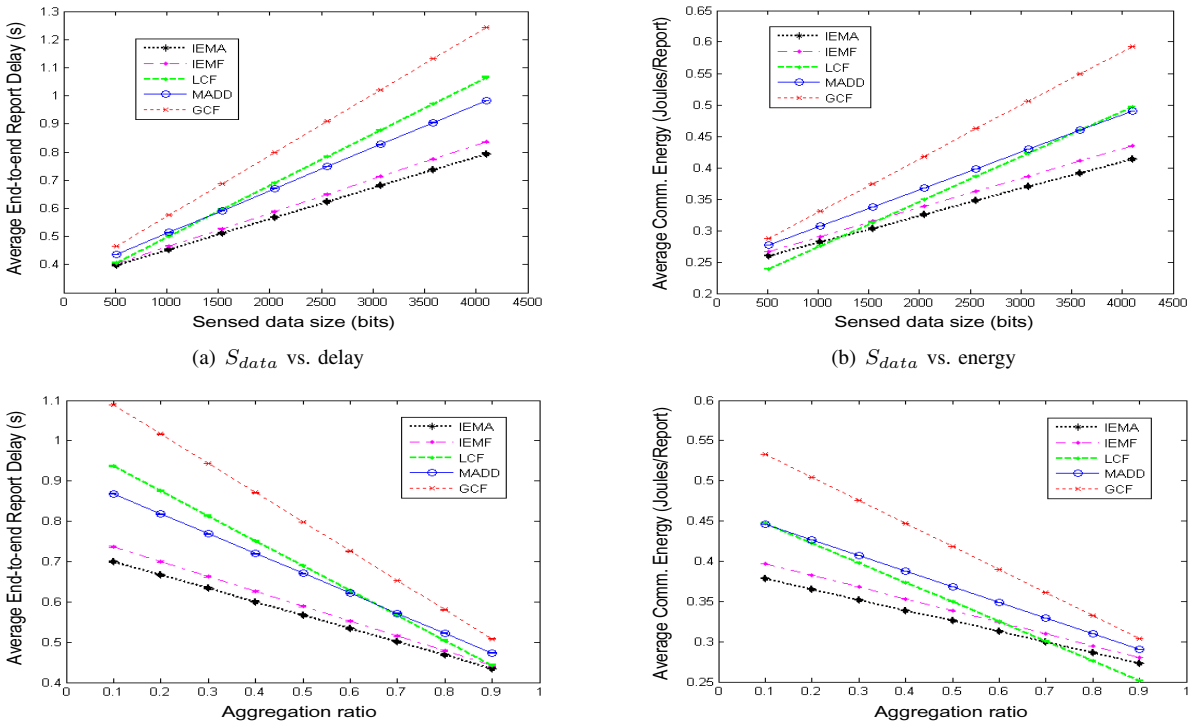
Fig. 2. The impact of κ on: (a) average end-to-end delay; (b) average communication energy.

fast convergence property of the proposed schemes are highly desirable.

We next examine the impact of several design parameters on the performance of the proposed algorithms, including sensor data size S_{data}, data aggregation ratio ρ, and the number of source nodes n. We also compare the proposed schemes with several representative schemes, including LCF [2], GCF [2], and MADD [4].

Figs. 3(a) and 3(b) are obtained with 15 source nodes, and by increasing the sensor data size S_{data} from 512 bits to 4096 bits. From the figures, it can be seen that IEMF and IEMA (with 15 iterations) achieve smaller delay and lower energy consumption than LCF, GCF, and MADD in most of the cases. More interestingly, the gap between the curves, i.e., the performance improvements achieved by IEMF and IEMA increases as S_{data} gets larger. The linear relationship between energy consumption and S_{data} is largely due to the energy consumption model adopted in the algorithm design= (see Section III-B), while delay is linear with S_{data} because the WSNs are generally lightly loaded.

In Figs. 3(c) and 3(d), we present the impact of the aggregation ratio ρ on the energy and delay performance, where ρ is increased from 0.1 to 0.9, representing different redundancy and compression schemes found in various applications. It can be seen that both IEMF and IEMA achieve smaller delays and energy consumption than LCF, GCF, and MADD in most of the cases. We also find that the IEMF curves are very close to the IEMA curves (i.e., the first iteration achieves the largest performance improvement). When ρ is increased close to 1, i.e., a nearly perfect aggregation scheme is used, the



(a) S_{data} vs. delay

(b) S_{data} vs. energy

(c) ρ vs. delay

(d) ρ vs. energy

Fig. 3. (a) & (b): the impact of sensed data size S_{data}. (c) & (d): the impact of Aggregation ratio ρ.

agent size is only increased slightly when it migrates along its itinerary. The room for optimization becomes smaller, and thus the improvements of IEMA are relatively smaller compared to the case of smaller ρ's.

We also observe considerable reductions in hop-count of the itinerary achieved by the proposed algorithms over the three existing algorithms. We omit the results due to lack of space.

ACKNOWLEDGMENT

This work was supported in part by the Canadian Natural Sciences and Engineering Research Council under grant STPGP 322208-05. Professor Shiwen Mao's work is supported in part by the US NSF under Grant ECCS-0802113 and through the Wireless Internet Center for Advanced Technology (WICAT) at Auburn University. This research was partially supported by KRCF and by the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center support program supervised by the IITA (grant number IITA-2009-C1090-0902-0006).

V. CONCLUSIONS

In this paper, we addressed the problem of optimized itinerary planning for MAs in dense WSNs. Based on a general data aggregation model, as well as relaxed assumptions, we presented IEMF, an extension of LCF that achieves improved energy performance by choosing the first source node to visit according to estimate communication costs. We also presented IEMA(κ), an iterative version of IEMF, where the selection of the first κ source nodes are optimized based on estimated energy costs. We showed that the proposed schemes achieve considerable improvements in both energy savings and delay over existing schemes, while IEMA(κ) provides a trade-off between energy cost and computational complexity. We will consider the more challenging case of itinerary planning for multiple MAs in our future work.

REFERENCES

[1] M. Chen, S. Gonzalez, and V.C.M. Leung, "Applications and design issues of mobile agents in wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp.20–26, Dec. 2007.

[2] H. Qi, and F. Wang, "Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks", in *Proc. IEEE ICC'01*, Helsinki, Finland, June 2001.

[3] Q. Wu, et al., "On computing mobile agent routes for data fusion in distributed sensor networks," *IEEE Trans. Knowledge and Data Engineering*, vol.16, no.6, pp.740–753, June 2004.

[4] M. Chen, T. Kwon, Y. Yuan, Y. Choi, and V.C.M. Leung, "Mobile agent-based directed diffusion in wireless sensor networks," *EURASIP Journal on Advances in Signal Processing*, 2007. DOI: 10.1155/2007/36871

[5] H. Qi, Y. Xu, X. Wang, "Mobile-Agent-Based Collaborative Signal and Information Processing in Sensor Networks," *Proceedings of the IEEE*, vol.91, no.8, pp.1172–1183, Aug. 2003.

[6] M. Chen, T. Kwon, Y. Yuan, and V.C.M. Leung ,"Mobile Agent Based Wireless Sensor Networks," *Journal of Computers*, vol.1, no.1, pp.14–21, Apr. 2006.

[7] Y. Tseng, S. Kuo, H. Lee, and C. Huang, "Location tracking in a wireless sensor network by mobile agents and its data fusion strategies," *The Computer Journal*, vol.47, no.4, pp.448-460, July 2004.

[8] Y. Xu and H. Qi, "Mobile agent migration modeling and design for target tracking in wireless sensor networks", *Ad Hoc Networks Journal*, vol. 6, no.1, pp.1-16, January 2008.

[9] A. Harris, R. Snader, and I. Gupta, "Building trees based on aggregation efficiency in sensor networks," *Ad Hoc Networks Journal*, vol.5, no.8, pp.1317–1328, Nov. 2007.

[10] C.Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. ACM MobiCom'00*, Boston, MA, August 2000.

[11] S. Mao and Y.T. Hou, "BeamStar: An edge-based approach to routing in wireless sensor networks," *IEEE Trans. Mobile Computing*, vol.6, no.11, pp.1284–1296, Nov. 2007.