Efficient Dispatch of Multi-Capability Mobile Sensors in Hybrid Wireless Sensor Networks

You-Chiun Wang
Department of Computer Science and Engineering,
National Sun Yat-sen University, Kaohsiung, 80424, Taiwan
Email: ycwang@cse.nsysu.edu.tw

***Abstract*—We consider a hybrid wireless sensor network consisting of static and mobile sensors, where each static sensor can detect an event of only one type while each mobile sensor can analyze events of multiple types. Static sensors monitor the environment and report where events appear in the sensing field. Then, mobile sensors are dispatched to reach these event locations to perform more in-depth analysis. An interesting problem is how to schedule mobile sensors' traveling paths so that their lifetime can be maximized. We thus formulate a multi-round multi-capability sensor dispatch problem, which is shown to be NP-hard. Through an economic viewpoint, we develop a heuristic using the Pareto optimality to solve this problem. Simulation results have verified the effectiveness of the proposed heuristic. The paper contributes in defining a new sensor dispatch problem and proposing efficient solution to it.**

***Index Terms*—energy saving, mobile sensor, Pareto optimality, sensor dispatch, wireless sensor network.**

I. Introduction

Wireless sensor networks (WSNs) enrich our life by providing context-aware monitoring of the physical environment. Hybrid WSNs consisting of static and mobile sensors possess more flexibility than conventional WSNs containing only static sensors [1], [2], [3]. On one hand, static sensors are used to conduct environmental sensing and maintain network communication. On the other hand, mobile sensors have more resources such as sensing capability, computing power, and energy. They can move to designated locations to carry out missions such as analyzing events or replacing broken nodes. Recently, hybrid WSNs have applications in [4], [5], [6], [7].

In this paper, we aim at the issue of dispatching "multi-capability" mobile sensors to the locations of events appearing in the sensing field. Static sensors will identify where suspicious events appear and report them to mobile sensors so as to carry out in-depth analysis. Consider a set of types $\mathcal{T}$ with which events may appear. Each event is associated with one type in $\mathcal{T}$. A sensor equipped with the sensing device used to detect events of type $t_i \in \mathcal{T}$ is said to have the *capability* t_i. We consider *single-capability static (SCS) sensors*, where each SCS sensor can detect an event of only one type. However, for events of each type in $\mathcal{T}$, the sensing field is deployed with sufficient SCS sensors to detect them. Therefore, SCS sensors can always identify where events appear. On the other hand, each mobile sensor is equipped with multiple sensing devices so that it can analyze events of a subset of types in $\mathcal{T}$. We call such sensors *multiple-capability mobile (MCM) sensors*. MCM sensors may have different capabilities. When an event of type $t_i \in \mathcal{T}$ appears, we can only dispatch an MCM sensor that has the capability t_i to analyze it. To ensure that we can find a feasible schedule to dispatch mobile sensors, the union of all MCM sensors' capabilities should be equal to $\mathcal{T}$.

The above scenario is practical in some long-term, multi-stage monitoring applications. For example, consider an air-pollution monitoring application as follows: In the initial stage, one may deploy static CO_2 (carbon dioxide) sensors to monitor the CO_2 concentration. Then, he/she may decide to add static SO_2 (sulphur dioxide) sensors to improve the monitoring result in the next stage. Thus, the WSN has two types of SCS sensors to report different events (for example, high CO_2 or SO_2 concentration). On the other hand, mobile sensors can be equipped with both CO_2 and SO_2 sensing devices to reduce the hardware cost. In this case, we have MCM sensors used to analyze events of multiple types.

Since MCM sensors are usually batter-powered and the moving cost dominates their energy consumption [8], [9], our objective is to emphasize the path efficiency of MCM sensors. Considering that events may appear anytime and anywhere, it would be inefficient to dispatch one mobile sensor immediately after an event appears. Therefore, we suggest dividing time into multiple rounds and schedule the traveling paths of MCM sensors in a round-by-round fashion. Then, we propose an *MCM sensor dispatch problem* which determines how to dispatch MCM sensors to the event locations given in each round such that the system lifetime can be maximized. Here, we define the system lifetime as the number of rounds until some event locations cannot be reached by any *suitable* MCM sensor due to lack of energy. In [10], it has been proven that even if all static and mobile sensors have a single capability (that is, the size of $\mathcal{T}$ is one), the problem of dispatching mobile sensors in a round-by-round fashion such that the system lifetime is maximum is still NP-complete. By letting each MCM sensor possess all capabilities in $\mathcal{T}$, we can reduce the above NP-complete problem to one of the instances of our MCM sensor dispatch problem, thereby showing it to be NP-hard.

At first glance, there seems to be a simple solution by handling event locations of each type in $\mathcal{T}$ separately. Specifically, we first divide event locations into groups according to their types. Then, for the group of each type $t_i \in \mathcal{T}$, we schedule MCM sensors to visit all of its event locations such that

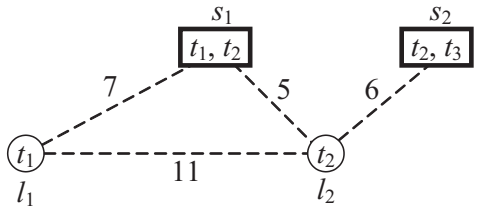

Fig. 1: An example of dispatching MCM sensors, where the number on each dotted line is the distance between two nodes.

these MCM sensors can consume the least amount of energy. Notice that only MCM sensors that have the capability t_i can participate in this schedule. The above procedure is repeated until all groups are handled. Then, for each MCM sensor, it can adopt any solution of the *traveling salesman problem (TSP)* to reach all event locations assigned to that MCM sensor. Fig. 1 gives an example, where $\mathcal{T} = \{t_1, t_2, t_3\}$ and we have two event locations l_1 and l_2 (represented by circles) and two MCM sensors s_1 and s_2 (represented by rectangles). The types of l_1 and l_2 are t_1 and t_2, respectively. On the other hand, s_1 and s_2 have capabilities $\{t_1, t_2\}$ and $\{t_2, t_3\}$, respectively. According to the above solution, we first schedule an MCM sensor to visit l_1. In this case, we choose s_1 since only it has the capability t_1. Then, for l_2, we still choose s_1 because it has a shorter moving distance to l_2 (compared with that of s_2). Thus, s_1 totally moves in a distance of 16 (= 5 + 11) while s_2 does not move.

The above solution adopts a divide-and-conquer concept, but it has two drawbacks. First, the overall moving distance of all MCM sensors is not reduced, even though we try to minimize the moving distance of MCM sensors for event locations of each type in $\mathcal{T}$. Second, some MCM sensors (for example, s_1) are burdened with heavy loads (that is, moving in a longer distance), leading them to exhaust energy quickly. In fact, there is a better solution to the example in Fig. 1, where s_1 and s_2 are dispatched to l_1 and l_2, respectively. In this case, the overall moving distance is reduced to 13 (= 7 + 6). Meanwhile, the loads of MCM sensors are balanced so as to extend the system lifetime [10]. Therefore, we should take all MCM sensors into account when determining their dispatch schedules, rather than just use a simple divide-and-conquer solution.

Motivated from the above observation, we propose an efficient solution to the MCM sensor dispatch problem by adopting the concept of *Pareto optimality*, which is widely used to address economic issues [11]. Given the event locations to be analyzed in each round, our solution calculates a matching between MCM sensors and event locations such that 1) the number of event locations paired by MCM sensors is maximum and 2) the matching is Pareto optimal. A matching $\mathcal{M}$ is Pareto optimal if there is no other matching $\mathcal{M}'$ such that no MCM sensor is worse off (that is, moves in a longer distance) in $\mathcal{M}'$ than in $\mathcal{M}$, and some MCM sensors are even better off (that is, move in shorter distances) in $\mathcal{M}'$ than in $\mathcal{M}$. In other words, we cannot swap two pairs of MCM sensors and event locations in $\mathcal{M}$ to reduce the overall moving distance of MCM sensors. Then, if there still exist unpaired event locations, we can repeat the above procedure until all event locations are assigned with MCM sensors. Compared with the simple divide-and-conquer solution, our MCM sensor dispatch solution takes all MCM sensors into consideration when calculating the dispatch schedule, thereby reducing their moving distances and extending the system lifetime.

In the literature, a large amount of research efforts [12], [13], [14] have elaborated on the issue of adopting mobile sensors to improve the sensing coverage of WSNs. The work of [15] suggests moving more sensors close to the locations of events predicted, while maintaining complete coverage of the sensing field. Considering that an object's trajectory may be predicted, [16] addresses how to maneuver mobile sensors to acquire data from that object in a real-time manner. The study in [17] uses static sensors to estimate coverage holes. Then, each mobile sensor selects the largest hole and moves to fill it.

Some variations of the sensor dispatch problem are also addressed. Given mobile sensors and target locations, [18] adopts a matching idea to dispatch mobile sensors to these target locations so that all mobile sensors can remain the maximum energy to conduct other missions (such as sensing or communication). On the other hand, [19] solves the similar problem in a distributed manner by letting mobile sensors compete to move to their nearest target locations. The above two studies aim at one-round sensor dispatch, while our paper considers multiple rounds of sensor dispatch. In [20], static sensors detecting events will invite mobile sensors to visit them to conduct more in-depth analysis. The mobile sensor which is closer and remains more energy, and whose leaving does not result in a big coverage hole, will be invited. Given event locations in each round, [10] tries to dispatch mobile sensors to visit all event locations such that their moving energy can be minimized and balanced, so the system lifetime can be prolonged. The above studies assume that all events have the same type, while our paper relaxes this assumption and allows events to have different types. The above features distinguish this paper from others.

We organize the paper as follows: Problem definition is given in Section II. Section III proposes our MCM sensor dispatch solution. Simulation results are presented in Section IV while Section V concludes this paper.

II. Problem Definition

Suppose that there is a set of types $\mathcal{T}$ of which events may appear in the sensing field. Each event is associated with exact one type in $\mathcal{T}$. We say that a sensor equipped with the sensing device which is able to detect events of type $t_i \in \mathcal{T}$ has the capability t_i. Consider a hybrid WSN deployed with both SCS and MCM sensors, where each SCS sensor has only one capability while each MCM sensor can have multiple capabilities. Sensors are assumed to know their own locations, which can be achieved by the global positioning system (GPS) or some other localization schemes [21]. For

each type $t_i \in \mathcal{T}$, t_i-capability SCS sensors are dense enough to fully cover the sensing field. In addition, all SCS sensors can form a connected network. Therefore, they can cooperate to identify events of different types which may appear in arbitrary locations in the sensing field. However, we make no assumption on the event distribution, and the occurrence of any two events is independent.

MCM sensors are more resource-rich and can be dispatched to event locations to conduct more in-depth analysis. Both the moving speed of an MCM sensor and its energy consumption to move a unit distance are considered as constants. In addition, the sensing field is assumed to be obstacle-free so that MCM sensors can move straight to their destinations. However, the capabilities of any two MCM sensors may not be necessarily the same. Therefore, when a location of t_i-type event is reported by SCS sensor(s), only those MCM sensors that have the capability t_i can be dispatched to that location. To guarantee that we can dispatch MCM sensors to analyze all possible events in the sensing field, the union of the capabilities of all MCM sensors should be equal to $\mathcal{T}$.

Since events may appear on arbitrary locations at any time, it would be efficient to gather the locations of events for a while and then schedule MCM sensors to move to these event locations. Therefore, we divide the time into multiple rounds. In each round, an event only needs to be analyzed by one MCM sensor. Our discussion thus aims at the dispatch problem in each round. Specifically, given a set of m event locations $\mathcal{L} = \{l_1, l_2, \cdots, l_m\}$ and a set of n MCM sensors $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$, where each location $l_i \in \mathcal{L}$ is associated with one type in $\mathcal{T}$, our objective is to calculate a dispatch schedule Φ_j for each $s_j \in \mathcal{S}$, which contains a sequence of event locations. Notice that s_j should have the corresponding capabilities to analyze all events in its Φ_j.

Let e_j be the current energy of an MCM sensor s_j and $\Phi_j(k)$ be the kth event location in s_j's dispatch schedule Φ_j. The energy required to complete s_j's dispatch schedule is

$$Energy(\Phi_j) = e_{\text{cost}} \times \left(d(s_j, \Phi_j(1)) + \sum_{k=1}^{|\Phi_j|-1} d(\Phi_j(k), \Phi_j(k+1)) \right),$$

where e_{cost} is the energy cost for an MCM sensor to move one unit distance, $|\Phi_j|$ is the number of event locations in Φ_j, and $d(\cdot,\cdot)$ is the distance between two locations. Obviously, any dispatch schedule Φ_j of each MCM sensor s_j should always satisfy that $e_j \geq Energy(\Phi_j)$. Assume that MCM sensors cannot be rechargeable. Given the initial energy of each MCM sensor in $\mathcal{S}$, the *MCM sensor dispatch problem* determines how to calculate the dispatch schedules of all MCM sensors in each round, such that the system lifetime, which is defined by the number of rounds until we cannot find any suitable MCM sensor to visit some event locations, can be maximized. Remark that when the remaining MCM sensors still have sufficient energy but they do not have a certain capability, say, $t_i \in \mathcal{T}$, if $\mathcal{L}$ has some event locations of type t_i, then we cannot dispatch any MCM sensor to visit these event locations and thus the system lifetime is terminated.

III. The Proposed Solution

We propose an MCM sensor dispatch algorithm, whose idea is to take all MCM sensors into account when scheduling them and to reduce their moving energy in each round. Without loss of generality, we remove those MCM sensors that do not have sufficient energy to reach any location in $\mathcal{L}$ from $\mathcal{S}$. Given $\mathcal{L}$ and $\mathcal{S}$ in each round, our MCM sensor dispatch algorithm has the following five steps:

- **Step 1:** Initially, we set $\Phi_j = \varnothing$ (that is, empty dispatch schedule) for each MCM sensor $s_j \in \mathcal{S}$.
- **Step 2:** From $\mathcal{L}$ and $\mathcal{S}$, we construct a weighted bipartite graph $\mathcal{G} = \{\mathcal{L} \cup \mathcal{S}, \mathcal{L} \times \mathcal{S}\}$. All event locations and all MCM sensors are converted into vertices. Edges only connect vertices between $\mathcal{L}$ and $\mathcal{S}$. In particular, an edge (l_i, s_j) exists if and only if event location l_i's type is $t_k \in \mathcal{T}$ and MCM sensor s_j has the capability t_k. In addition, we also associate edge (l_i, s_j) with a weight
$$w(l_i, s_j) = e_{\text{cost}} \times d(l_i, s_j),$$
to indicate the energy consumption for s_j to move from its current location to the event location l_i.
- **Step 3:** We then calculate a maximum Pareto optimal matching $\mathcal{M}$ from graph $\mathcal{G}$. Obviously, $\mathcal{M}$ must be in the form of pairs of event locations and MCM sensors (for example, $\mathcal{M} = \{(l_{\alpha_1}, s_{\beta_1}), (l_{\alpha_2}, s_{\beta_2}), \cdots, (l_{\alpha_k}, s_{\beta_k})\}$, where $k \leq \min(m, n)$). Then, for each MCM sensor s_j in $\mathcal{M}$, we insert its paired event location into its dispatch schedule Φ_j. We then remove each event location $l_i \in \mathcal{M}$ from $\mathcal{L}$ because l_i has been assigned with one MCM sensor to analyze it.
- **Step 4:** Repeat steps 2 and 3 until $\mathcal{L}$ becomes empty (in other words, all event locations have been assigned with MCM sensors to analyze them).
- **Step 5:** For each MCM sensor s_j, we can adopt any TSP (approximation) solution to arrange all event locations in its dispatch schedule Φ_j so that s_j can consume the minimum amount of energy to complete the current dispatch schedule.

The remaining issue in the aforementioned algorithm is how to calculate a maximum Pareto optimal matching in step 3. To calculate such a matching, we associate each event location $l_i \in \mathcal{L}$ with a *preference list* which ranks all of the MCM sensors acceptable to l_i (in other words, these MCM sensors have the corresponding capability to analyze l_i). The rank of an edge (l_i, s_j) is thus the rank that event location l_i has assigned to MCM sensor s_j. We define that an event location l_i "prefers" another matching $\mathcal{M}_2$ to the matching $\mathcal{M}_1$ if any one of the following two conditions is satisfied:

1) Event location l_i is paired with some MCM sensor in $\mathcal{M}_2$ but not in $\mathcal{M}_1$.
2) Suppose that l_i is paired with MCM sensors s_j and s_k in $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. We have $w(l_i, s_k) <$

$w(l_i, s_j)$. In other words, s_k can consume less energy to reach l_i compared with s_j (and therefore l_i "likes" s_k better than s_j).

However, if an event location is not paired with any MCM sensor in both matchings, it is considered as *no difference* between matchings $\mathcal{M}_1$ and $\mathcal{M}_2$. According to the above definition, we say that a matching $\mathcal{M}$ is *Pareto optimal* if there is no other matching $\mathcal{M}'$ such that 1) at least one event location prefers the MCM sensor paired in $\mathcal{M}'$ to the MCM sensor paired in $\mathcal{M}$ and 2) no event location prefers the MCM sensor paired in $\mathcal{M}$ to the MCM sensor paired in $\mathcal{M}'$. Notice that given a weighted bipartite graph, it is possible to find multiple Pareto optimal matchings with different sizes [22]. Therefore, in order to increase the calculation efficiency in step 3 of our algorithm, we need to find the Pareto optimal matching with the maximum size (that is, the number of paired event locations and MCM sensors is maximum).

Based on [23], a three-step method to calculate the maximum Pareto optimal matching $\mathcal{M}$ from graph $\mathcal{G}$ is developed as follows:

a) Given $\mathcal{G}$, we first calculate a maximum matching $\mathcal{M}$. This can be achieved by the Hopcroft-Karp algorithm proposed in [24].
b) For each event location l_i in $\mathcal{M}$, we conduct the following check: Suppose that l_i is currently paired with MCM sensor s_j in $\mathcal{M}$. If there is an *unpaired* MCM sensor, say, s_k such that $w(l_i, s_j) > w(l_i, s_k)$, we remove pair (l_i, s_j) from $\mathcal{M}$ and then add pair (l_i, s_k) to $\mathcal{M}$. When there are multiple candidates, we select the MCM sensor s_k such that $w(l_i, s_k)$ is minimum.
c) We then check whether event locations can "trade" with their currently paired MCM sensors in $\mathcal{M}$ so that the overall energy consumption of MCM sensors can be further reduced. In particular, if $\mathcal{M}$ contains a sequence of pairs $(l_{\alpha_1}, s_{\beta_1}), (l_{\alpha_2}, s_{\beta_2}), \cdots$, and $(l_{\alpha_k}, s_{\beta_k})$ such that $w(l_{\alpha_1}, s_{\beta_1}) > w(l_{\alpha_1}, s_{\beta_2}), w(l_{\alpha_2}, s_{\beta_2}) > w(l_{\alpha_2}, s_{\beta_3}), \cdots, w(l_{\alpha_{k-1}}, s_{\beta_{k-1}}) > w(l_{\alpha_{k-1}}, s_{\beta_k})$, and $w(l_{\alpha_k}, s_{\beta_k}) > w(l_{\alpha_k}, s_{\beta_1})$, we remove pairs $(l_{\alpha_1}, s_{\beta_1})$, $(l_{\alpha_2}, s_{\beta_2})$, $\cdots$, and $(l_{\alpha_k}, s_{\beta_k})$ from $\mathcal{M}$ and then add pairs $(l_{\alpha_1}, s_{\beta_2})$, $(l_{\alpha_2}, s_{\beta_3})$, $\cdots$, $(l_{\alpha_{k-1}}, s_{\beta_k})$, and $(l_{\alpha_k}, s_{\beta_1})$ to $\mathcal{M}$.

In the above method, since step a) only finds a matching that has the maximum size (without considering the weights of the selected edges), we need additional two steps to "refine" the matching. Specifically, in step b), although an event location has already been assigned with one MCM sensor, the event location can still test if there are *free* MCM sensors (that is, these MCM sensors are not selected in $\mathcal{M}$) which are closer to it. If so, the event location can select the closest free MCM sensor to reduce the overall energy consumption of MCM sensors. On the other hand, step c) allows event locations to exchange their currently assigned MCM sensors to further reduce the overall energy consumption. Specifically, if there is a sequence of event locations $(l_{\alpha_1}, l_{\alpha_2}, \cdots, l_{\alpha_k})$ in $\mathcal{M}$ such that each event location l_{α_p} prefers the MCM sensor currently paired by $l_{\alpha_{p+1}}$ (that is, $s_{\beta_{p+1}}$) to its currently paired MCM sensor (that is, s_{β_p}), l_{α_p} and $l_{\alpha_{p+1}}$ can make a trade by exchanging their assigned MCM sensors. In this way, MCM sensors can move in a shorter distance and therefore reduce their moving energy consumption.

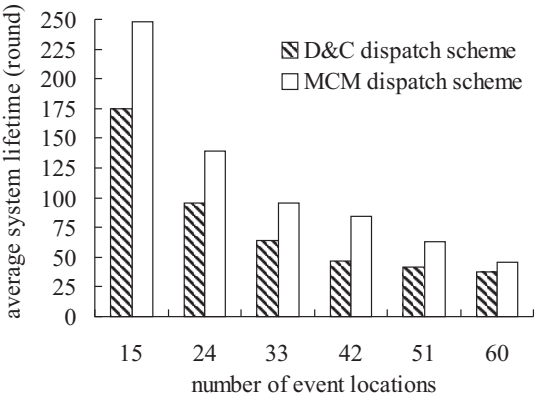

Fig. 2: Comparison on the average system lifetime.

IV. Simulation Results

We have developed a simulator using the C++ language to evaluate the performance of our dispatch scheme. In our simulations, the sensing field is a 550 meters $\times$ 450 meters rectangle. Approximately 1000 static sensors are deployed in a hexagon-like manner [25]. We set the sensing distance and the communication distance of each static sensor to 10 meters and 17.5 meters, respectively, so that all static sensors can form a connected network. In each experiment, we randomly select a subset of static sensors to be event locations (that is, $\mathcal{L}$). In addition, we set $\mathcal{T} = \{t_1, t_2, t_3\}$. For each type $t_i \in \mathcal{T}$, a third of event locations are associated with t_i. On the other hand, the number of MCM sensors is 20 in each experiment, and they have an initial energy of 10000 units used for movement. When an MCM sensor moves one unit distance, it will spend one unit of energy (that is, $e_{\text{cost}} = 1$). Among these 20 MCM sensors, five MCM sensors have the capabilities $\{t_1, t_2\}$, five MCM sensors have the capabilities $\{t_2, t_3\}$, five MCM sensors have the capabilities $\{t_1, t_3\}$, and the remaining five MCM sensors have all capabilities $\{t_1, t_2, t_3\}$. We compare our proposed scheme in Section III (denoted by "MCM dispatch scheme") with the divide-and-conquer scheme mentioned in Section I (denoted by "D&C dispatch scheme").

We first investigate the average system lifetime under different dispatch schemes, as shown in Fig. 2. The number of event locations is set to 15, 24, 33, 42, 51, and 60. When the number of event locations grows from 15 to 24, the system lifetime substantially reduces because the number of MCM sensors becomes smaller than the number of event locations. In this case, some MCM sensors have to take care of more than one event location in every round, thereby exhausting their energy quickly. By adopting the Pareto optimality and taking all MCM sensors into account when determining their dispatch schedules, our MCM dispatch scheme can significantly save

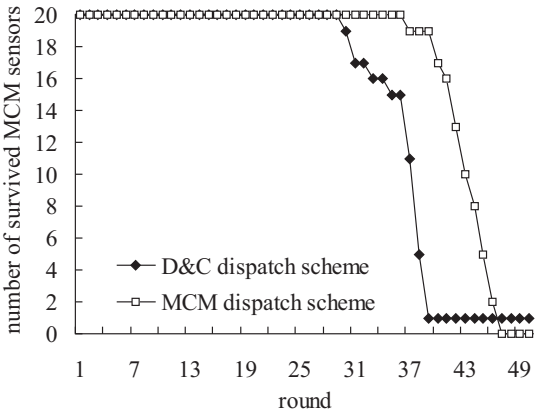

Fig. 3: Comparison on the number of survived MCM sensors.

the moving energy of MCM sensors, thereby extending the system lifetime. Observing from Fig. 2, our MCM dispatch scheme can extend approximately 47.71% of the system lifetime compared with the D&C dispatch scheme in this simulation.

We then measure the number of survived MCM sensors, as shown in Fig. 3. The number of event locations is set to 60. We observe that the first MCM sensor exhausts its energy at the 30th and 37th rounds under the D&C dispatch scheme and the MCM dispatch scheme, respectively. After the first MCM sensor dies, other MCM sensors are burdened with more loads so that they will exhaust their energy soon. However, our MCM dispatch scheme can postpone such a phenomenon, which demonstrates its effectiveness. Interestingly, the D&C dispatch scheme terminates its system lifetime at the 38th round, but one MCM sensor still survives. This phenomenon shows that the D&C dispatch scheme burdens MCM sensors with imbalanced loads, which hurts the system lifetime.

V. Conclusions

In this paper, we have developed an efficient algorithm to dispatch MCM sensors in a hybrid WSN from an economic viewpoint. We formulate an MCM sensor dispatch problem and show it to be NP-hard. By constructing a weighted bipartite graph, we translate the MCM sensor dispatch problem to a matching problem and then calculate the dispatch schedules of MCM sensors by finding the maximum Pareto optimal matching. Simulation results have shown that our proposed dispatch solution can significantly extend the system lifetime compared with the simple divide-and-conquer solution.

Acknowledgement

You-Chiun Wang's research is co-sponsored by the National Science Council under grant no. 100-2218-E-110-006-MY3, Taiwan.

References

[1] G. Cao, G. Kesidis, T. F. L. Porta, B. Yao, and S. Phoha, "Purposeful mobility in tactical sensor networks," *Sensor Network Operations*, 2006.

[2] Y. C. Wang and Y. C. Tseng, "Intentional mobility in wireless sensor networks," *Wireless Networks: Research, Technology and Applications*, 2009.

[3] Y. C. Wang, F. J. Wu, and Y. C. Tseng, "Mobility management algorithms and applications for mobile sensor networks," *Wireless Communications and Mobile Computing*, vol. 12, no. 1, pp. 7–21, 2012.

[4] T. Wark, P. Corke, P. Sikka, L. Klingbeil, G. Ying, C. Crossman, P. Valencia, D. Swain, and G. Bishop-Hurley, "Transforming agriculture through pervasive wireless sensor networks," *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 50–57, 2007.

[5] Y. C. Tseng, Y. C. Wang, K. Y. Cheng, and Y. Y. Hsieh, "iMouse: an integrated mobile surveillance and wireless sensor system," *Computer*, vol. 40, no. 6, pp. 60–66, 2007.

[6] J. Lu and T. Suda, "Differentiated surveillance for static and random mobile sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, pp. 4411–4423, 2008.

[7] S. C. Hu, Y. C. Wang, C. Y. Huang, and Y. C. Tseng, "Measuring air quality in city areas by vehicular wireless sensor networks," *Journal of Systems and Software*, vol. 84, no. 11, pp. 2005–2012, 2011.

[8] R. Rao and G. Kesidis, "Purposeful mobility for relaying and surveillance in mobile ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 225–231, 2004.

[9] Y. C. Wang, W. C. Peng, M. H. Chang, and Y. C. Tseng, "Exploring load-balance to dispatch mobile sensors in wireless sensor networks," *Proc. IEEE International Conference on Computer Communications and Networks*, pp. 669–674, 2007.

[10] Y. C. Wang, W. C. Peng, and Y. C. Tseng, "Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 12, pp. 1836–1850, 2010.

[11] A. Abdulkadiroglu and T. Sonmez, "Random serial dictatorship and the core from random endowments in house allocation problems," *Econometrica*, vol. 66, no. 3, pp. 689–701, 1998.

[12] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man and Cybernetics–Part A*, vol. 35, no. 1, pp. 78–92, 2005.

[13] T. M. Cheng and A. V. Savkin, "A distributed self-deployment algorithm for the coverage of mobile wireless sensor networks," *IEEE Communications Letters*, vol. 13, no. 11, pp. 877–879, 2009.

[14] N. Bartolini, T. Calamoneri, T. F. L. Porta, and S. Silvestri, "Autonomous deployment of heterogeneous mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 10, no. 6, pp. 753–766, 2011.

[15] Z. Butler and D. Rus, "Event-based motion control for mobile-sensor networks," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 34–42, 2003.

[16] M. D. Naish, E. A. Croft, and B. Benhabib, "Dynamic dispatching of coordinated sensors," *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3318–3323, 2000.

[17] G. Wang, G. Cao, and T. L. Porta, "Selection and navigation of mobile sensor nodes using a sensor network," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 563–576, 2007.

[18] Y. C. Wang, C. C. Hu, and Y. C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Transactions on Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.

[19] Y. C. Wang and Y. C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.

[20] A. Verma, H. Sawant, and J. Tan, "Selection and navigation of mobile sensor nodes using a sensor network," *Pervasive and Mobile Computing*, vol. 2, no. 1, pp. 65–84, 2006.

[21] H. C. Chu and R. H. Jan, "A GPS-less self-positioning method for sensor networks," *Proc. IEEE International Conference on Parallel and Distributed Systems*, pp. 629–633, 2005.

[22] R. Fleischer and Y. Wang, "Dynamic Pareto optimal matching," *Proc. IEEE International Symposium on Information Science and Engineering*, pp. 797–802, 2008.

[23] D. J. Abraham, K. Cechlarova, D. F. Manlove, and K. Mehlhorn, "Pareto optimality in house allocation problems," *Proc. International Conference on Algorithms and Computation*, pp. 1163–1175, 2005.

[24] J. E. Hopcroft and R. M. Karp, "A $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM Journal on Computing*, vol. 2, pp. 225–231, 1973.

[25] Y. C. Wang, C. C. Hu, and Y. C. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," *Proc. IEEE Wireless Internet Conference*, pp. 114–121, 2005.