Node Reclamation and Replacement for Long-Lived Sensor Networks

Bin Tong, *Member*, *IEEE*, Guiling (Grace) Wang, *Member*, *IEEE*, Wensheng Zhang, *Member*, *IEEE*, and Chuang Wang

Abstract—When deployed for long-term tasks, the energy required to support sensor nodes' activities is far more than the energy that can be preloaded in their batteries. No matter how the battery energy is conserved, once the energy is used up, the network life terminates. Therefore, guaranteeing long-term energy supply has persisted as a big challenge. To address this problem, we propose a *node reclamation and replacement (NRR) strategy*, with which a mobile robot or human labor called *mobile repairman* (MR) periodically traverses the sensor network, reclaims nodes with low or no power supply, replaces them with fully charged ones, and brings the reclaimed nodes back to an energy station for recharging. To effectively and efficiently realize the strategy, we develop an *adaptive rendezvous-based two-tier scheduling scheme (ARTS)* to schedule the replacement/reclamation activities of the MR and the duty cycles of nodes. Extensive simulations have been conducted to verify the effectiveness and efficiency of the ARTS scheme.

Index Terms—Sensor networks, node reclamation and replacement, energy replenishment, duty-cycle scheduling.

1 Introduction

In a wireless sensor network (WSN), sensor nodes are powered by batteries that can operate for only a short period of time, which results in short network lifetime. The short lifetime disables the application of WSNs for long-term tasks such as structural health monitoring for bridges and tunnels, border surveillance, road condition monitoring, and so on. Hence, many energy conservation schemes were proposed to battle the constraint. With these schemes, the rate of energy consumption is slowed down, but consumed energy cannot be compensated. Therefore, the effectiveness of these schemes is inherently restrained by the amount of energy preloaded to sensor nodes.

Fully addressing the problem requires that energy be continually replenished to sensor nodes. One possible approach is to harvest energy from various environmental sources [1], [2], [3], [4], [5] such as the sunlight. However, efficient harvesting technologies are still absent. In particular, the amount of energy that a solar cell can harvest is proportional to its surface area, but it is infeasible to equip a tiny sensor node with a large-size solar cell. The amount of available solar energy also depends on uncontrollable conditions such as cloudiness of the sky. Hence, it is very likely that the energy harvested is limited and unable to satisfy the needs of sensor nodes. Another solution is to incrementally deploy new sensor nodes to take over sensor nodes running out of energy. This approach, however, is costly because sensor node hardware cannot be reused, and more importantly, it causes pollution to the environment because dead batteries and hardware are left in the environment. Seeking an effective and efficient way to guarantee long-term energy supply remains as a big challenge.

Revisiting the problem from a different angle, we propose a new strategy called *node reclamation and replacement (NRR)*. With the NRR strategy, a robot or human labor called *mobile repairman (MR)* periodically reclaims sensor nodes of low or no energy supply, replaces them with fully charged sensor nodes, and brings the reclaimed sensor nodes back to a place called *energy station (ES)*; in the ES, the reclaimed sensor nodes are recharged, temporarily stored, and can be used to replace other sensor nodes in later time. This approach is applicable to WSNs that are deployed in environments accessible to robots or human labors, such as roadsides, factories, parks, forests, etc.

The basic idea of the NRR strategy may appear simple, but effective and efficient realization of the strategy is challenging. Ideally, the NRR scheme should schedule both the travels of the MR and the duty cycles of sensor nodes to achieve the following goals simultaneously: first, *a guaranteed quality of service should be provided*. The duty cycles of sensor nodes should be properly scheduled to ensure a sufficient number of sensor nodes being alive before the MR's visit. Moreover, the number of sensor nodes needed to be reclaimed/replaced each time should be small such that the MR is able to complete reclamation and replacement in time considering the limited number of sensor nodes that the MR can carry at one time. This implies that sensor nodes should not die around the same time, and therefore, the widely practiced load balancing philosophy and techniques do not apply. Second, *the overhead caused by the reclamation and replacement should be minimized*. The MR's travel should be properly scheduled such that the travel distance of the MR is minimized in the long run. What is even more

- *B. Tong is with the Microsoft Corporation, One Microsoft Way, Redmond, WA 98052. E-mail: tongbin@iastate.edu.*
- *G. (Grace) Wang is with the Department of Computer Science, New Jersey Institute of Technology, University Heights, Newark, NJ 07102. E-mail: gwang@njit.edu.*
- *W. Zhang and C. Wang are with the Department of Computer Science, Iowa State University, 226 Atanasoff Hall, Ames, IA 50011-1041. E-mail: wzhang@iastate.edu, cwang@cs.iastate.edu.*

Manuscript received 23 Dec. 2009; revised 23 Apr. 2010; accepted 22 June 2010; published online 5 Jan. 2011.
Recommended for acceptance by A. Nayak.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2009-12-0662.
Digital Object Identifier no. 10.1109/TPDS.2011.25.

challenging is that *the travel scheduling of the MR and the duty cycling of sensor nodes are tightly coupled*. How sensor nodes determine their duty cycles locally depends on when the MR comes to replace them; meanwhile, the travel schedule of the MR depends on the energy level of sensor nodes.

We propose an *adaptive rendezvous-based two-tier scheduling scheme (ARTS)* to tackle the above problem and thus realize the NRR strategy effectively and efficiently. In the scheme, sensor node reclamation and replacement are performed round by round. During each round, scheduling is conducted in two tiers: the global tier and the local tier. The global-tier scheduling determines how many sensor nodes should be reclaimed and replaced as well as in what order the MR should reclaim and replace these sensor nodes. Meanwhile, sensor nodes in the network collaborate to conduct local-tier scheduling to determine their duty cycles. The two tiers of scheduling interact with each other through certain visiting appointments (*rendezvous*) that can be changed from round to round adaptively. In each round, the scheduling aims at 1) maintaining required quality of service, 2) concentrating energy consumption at sensor nodes that are to be reclaimed and replaced (hence, the amount of energy remaining in these sensor nodes is minimized when they are reclaimed), and 3) reducing the travel cost of the MR. This way, high efficiency of reclamation and replacement is achieved in each round, eventually leading to high efficiency in the long run. Extensive simulations are conducted to evaluate the proposed scheme. The results show that the ARTS scheme meets the objectives of the NRR strategy. The results also provide insights for network designers to choose appropriate system parameters when the ARTS scheme is deployed.

In the rest of the paper, Section 2 presents system model. An overview of the ARTS scheme is presented in Section 3. Sections 4 and 5 describe the local-tier scheduling and global-tier scheduling schemes in detail. Section 6 reports the simulation results. Section 7 discusses some practical issues in realizing the ARTS scheme. Section 8 discusses related work and Section 9 concludes the paper.

2 System Model

We consider a WSN that is deployed for long-term surveillance. The network needs to monitor a number of locations, called *posts*. Surrounding each post, a number of sensors are deployed, and these sensors form a *group*. The sensors in a group are close to each other, and thus they can provide approximately the same sensing quality. For example, for event detection, the sensing quality of two homogeneous sensors located close to each other is similar as these sensors have the similar probability to sense an event occurring in their sensing range.

Deploying multiple sensors near the same location is necessary for a number of applications. For example, in order to control sensing sensitivity [6] or improve accuracy of event detection, the readings from multiple sensors monitoring the same phenomenon can be collected and processed; relying on the readings from a single sensor may not achieve the required sensitivity or accuracy.

We assume that the locations of posts are given. In practice, they may be determined based on the shape of the deployment field, the required sensing quality, and the sensing range of each sensor. Our system model applies in the case where discrete points of interests in the sensing field need to be monitored, while does not apply in the case where the whole field needs to be monitored.

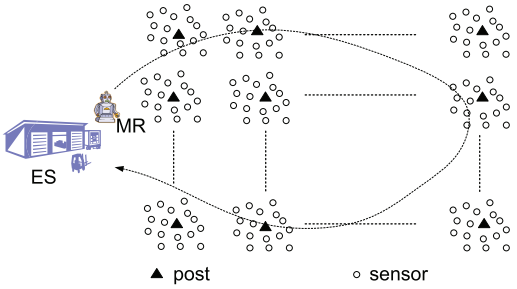

Fig. 1. System architecture. Sensors surrounding each post are very close to each other.

The system architecture of the NRR strategy is shown in Fig. 1. As can be seen, the system consists of a *mobile repairman*, an *Energy Station*, and a WSN composed of groups of sensors surrounding the posts. The MR traverses the network periodically to reclaim sensors having low or no energy, and replace them with fully charged sensors. The NRR strategy has the following assumptions:

- All sensors are loosely time synchronized. Existing time synchronization protocols [7], [8] can be used for this purpose. Time is divided into *phases*, of a constant length. A certain number of phases compose a *round*, the length of which is denoted as l. The MR visits each post at most once every round.
- A sensor has two modes: *active* and *sleeping*. For every phase, if a sensor is in the active mode, its energy is reduced by δ; if it is in the sleeping mode, its energy is unchanged. Let the energy of a fully charged sensor be e, which is a multiple of δ. If a sensor is in the active mode all the time, its lifetime is denoted as τ.
- Each group is deployed with N_d sensors, and N_d is called *group size*. At the beginning of each phase, all sensors in each group should wake up and participate in the duty-cycle scheduling. A sensor active in the previous phase collects sensory readings from all the other sensors also active in the previous phase, and based on these readings, it determines the number of sensors that need to be active in the current phase. We call this number *surveillance number*. Surveillance number varies between N_{min} and N_{max}. The number may be determined based on if there are events taking place at the post monitored by the group and other factors. The sensor announces surveillance number, and all the other sensors listen to the announcement and decide whether they will be in the active mode or in the sleeping mode in the current phase.
- The MR has a limited capacity, denoted as C, which is defined as the maximum number of (reclaimed or fully recharged) sensors it can carry. The MR has orientation and localization ability such that it can travel to designated locales and perform sensor replacement task.

TABLE 1
Notations

Notation	*Definition*
N_{max}	upper bound of surveillance number
N_{min}	lower bound of surveillance number
C	capacity of the MR
l	length of round
δ	amount of energy consumed by an active sensor in a phase
τ	lifetime of a sensor if it is active all the time
e	full amount of energy of a sensor
$N_r(i,j)$	replacement number at group g_i in round j
M	number of physical tours in a super tour

Note that the NRR strategy does not require a WSN to be connected. If a WSN is not connected, it often implies that delay in collecting sensory data can be tolerated. In this case, the MR can also serve as the mobile data collector. No matter the network is connected or not, the NRR strategy does not require each group to report its energy status through multi-hop communication. Instead, the MR collects the energy status of each group only when it visits the group. This way, communication overhead can be reduced.

In the following sections, a number of notations are to be used, and we summarize them in Table 1.

3 Overview of the ARTS Scheme

The proposed adaptive rendezvous-based two-tier scheduling scheme runs round by round. In each round, the mobile repairman visits each group at most once. When the MR visits a group, it reclaims/replaces a number of sensors, collects the information about residual energy in the group, and notifies the group of its next visiting time as well as the number of sensors to be reclaimed/replaced in the next visit. The calculation of the travel schedule and the number of sensors to be reclaimed/replaced are referred to as *global-tier scheduling*. Provided the MR's visiting time, sensors in the group collaborate in scheduling their activities, which is referred to as *local-tier scheduling*.

When the MR visits a group in each round j, it reclaims/replaces sensors, collects information about the residual energy of the group, and notifies its visiting time and the number of sensors to be reclaimed/replaced at round $j+1$. After the MR visits all groups, at the end of round j, the MR knows the residual energy of all the groups. Based on this information, the MR employs our proposed global-tier scheduling algorithm to calculate its visit time and the number of sensors to be reclaimed/replaced for each group in round $j+2$. Then, in round $j+1$, the MR notifies the groups, its schedule for round $j+2$ and meanwhile, collects information for the global-tier scheduling for round $j+3$.

One fundamental objective of both global-tier scheduling and local-tier scheduling is that the quality of service will not be violated, i.e., there are always enough sensors alive and working. For the global-tier scheduling, this means the MR cannot visit the groups too late; otherwise, there are not enough sensors alive before the MR comes. For the local-tier scheduling, given the MR's next visiting time, it must schedule the tasks among sensors effectively such that there are enough sensors alive before the MR visits.

With the above fundamental objective as a prerequisite, there is another objective for the scheduling in both tiers: the amount of remaining energy in sensors to be reclaimed/replaced should be as small as possible. Note that, by overusing the sensors to be reclaimed/replaced, the energy of other sensors would remain high and thus reduce the workload for further reclamation/replacement. With this objective, the local-tier scheduling algorithm becomes fundamentally different from most of the existing scheduling algorithms that are targeted at load balancing; instead, it should overuse some sensors. For the global-tier scheduling, this objective means the MR should not visit the groups too early; otherwise, the sensors to be reclaimed still have a lot of energy.

In addition, especially for the global-tier scheduling, the travel distance of the MR should be minimized to save both time and energy of the MR. This objective together with the above objectives makes the travel scheduling for the MR a NP-complete problem. We formally prove its NP-completeness and propose efficient heuristic solutions in Section 5 after we present the local-tier scheduling algorithm in the next section.

4 Local-Tier Scheduling

Local-tier scheduling is performed every phase in each group. All sensors in a group have a consistent view regarding the amount of remaining energy in all sensors in the group. This is because each sensor runs the same local-tier scheduling algorithm, thus it knows which sensors shall be in the active mode for any phase; when the MR visits the group, it will notify the group the number of sensors to be reclaimed/replaced in the next round, thus all sensors know which sensors are to be reclaimed/replaced based on the agreement that the least energy supplied sensors will be reclaimed/replaced. It is possible that node failures cause inconsistency temporarily in the view regarding the amount of remaining energy in all sensors. We provide a solution to it and discuss the solution in Section 7.

At the beginning of each phase, all sensors should wake up and participate in the scheduling. For each group, we choose a *leading sensor* out of the sensors active in the previous phase. The leading sensor, for example, could be the one with the smallest ID among active sensors. It collects sensory readings from other sensors active in the previous phase. Based on these readings, the leading sensor determines the surveillance number (the required number of active sensors) in the current phase, and announces the number to all sensors in the group. How the surveillance number is determined depends on application requirements and is out of the scope of this paper. Then, each sensor in the group runs our proposed local-tier scheduling algorithm independently to determine which sensors shall be in the active mode. If a sensor should not be active, it will go back to sleep. Our algorithm ensures that, if sensors have consistent view regarding the amount of remaining energy in all sensors in the group, they can reach the same scheduling decision.

Alternatively, we can let the leading sensor run the duty-cycle scheduling algorithm, and broadcast the result to the other sensors in the group. The result is encapsulated in a bit vector in which each bit represents a sensor. If a sensor's corresponding bit is 1, it should be active in the phase; otherwise, it should be sleeping in the phase. When the number of sensors in a group is small, the communication overhead for broadcasting this vector is comparable to that for broadcasting the surveillance number considering the header and trailer of a packet.

In either method described above, assuming all sensors in a group can reach each other in one hop, the communication complexity is $N_d + 1$ messages for each group in one phase, where N_d is group size.

The local-tier scheduling algorithm has two objectives: 1) quality of service is guaranteed, i.e., there are always at least N_{max} sensors alive in the group for any phase before the MR's next visit to the group (note that without knowing future energy consumption pattern, we have to assume the worst case, i.e., using N_{max} as surveillance number). 2) With objective 1 as a prerequisite, the local-tier scheduling should be conducted such that the residual energy in sensors to be reclaimed/replaced is minimized.

The inputs to the local-tier scheduling algorithm include the number of sensors in the group (denoted as N_d) and their residual energy, required surveillance number for the current phase, and the number of remaining phases before the MR's next visit. The output of the algorithm is a set of sensors that should be active for the current phase. The schedule must guarantee that the number of alive sensors is no less than N_{max} for all the subsequent phases before the MR's next visit.

We present the local-tier scheduling algorithm in three steps: we first introduce a *guard inequality*, satisfying which objective 1 can be attained. Then, we present a *controlled-greedy* algorithm which always attempts to schedule first the sensors with the lowest energy supply if doing so does not violate the guard inequality, and these sensors will be chosen to be reclaimed/replaced when the MR comes. Finally, we formally prove that the local-tier scheduling algorithm does not violate the guard inequality and thus guarantees the quality of surveillance service.

4.1 A Guard Inequality

To guarantee quality of service, we discover a guard inequality which has an attractive property that, for any phase, satisfying the inequality guarantees that we can always find a duty-cycle schedule such that at least N_{max} sensors are alive for the current phase and all the subsequent phases before the MR's next visit. Before introducing the inequality, we first introduce the following data structures: for each sensor, its ID is denoted as u_i $(0 \le i \le N_d - 1)$, and the amount of its remaining energy is denoted as e_i. The sensors are sorted into a list L according to the *decreasing* order of 2-tuples $\langle e_i, u_i \rangle$. In our algorithm, we assume the full energy of a sensor e is an integral multiple of δ. In reality, e may not be an integral multiple of δ. For example, $e = q\delta + r$, where q is an integer, and $r < \delta$. In this case, we treat the full energy of a sensor as $q\delta$.

Let t be the number of remaining phases before the next replacement/reclamation and δ be the amount of energy consumed by an active sensor per phase. We divide L into three sublists as follows:

- $L_1 = \langle u_0, \ldots, u_{m-1} \rangle$. Each sensor in L_1 has remaining energy of at least $t\delta$.
- $L_2 = \langle u_m, \ldots, u_{N_a-1} \rangle$. Each sensor in L_2 has remaining energy of less than $t\delta$, but at least δ.
- $L_3 = \langle u_{N_a}, \ldots, u_{N_d-1} \rangle$. Each sensor in L_3 has 0 remaining energy.

We call m the *turning* point. Sensors in sublists L_1 and L_2 are alive sensors, while sensors in sublist L_3 are dead sensors. The following is the theorem that formally introduces the guard inequality:

Theorem 4.1. *If Inequality (1) is satisfied at the beginning of a phase, a duty-cycle schedule can be found for the current phase and all the subsequent phases before the MR's next visit, such that the quality of service is guaranteed for these phases.*

$$\sum_{i=m}^{N_a-1} e_i \ge (N_{max} - m)t\delta. \tag{1}$$

Inequality (1) is called the guard inequality.

Proof (Sketch). We first prove that if the guard inequality is satisfied at the beginning of any phase, the quality of service is satisfied for that phase. We need to show that the number of alive sensors N_a is at least N_{max}. We consider two cases:

- $m >= N_{max}$. Clearly, there are more alive sensors than N_{max}.
- $m < N_{max}$. Suppose at the beginning of a phase, Inequality (1) holds. Each sensor u_i $(m \le i \le N_a - 1)$ belongs to L_2, and its remaining energy e_i is subject to $e_i < t\delta$. Hence, it follows $N_a - m > N_{max} - m$, i.e., $N_a > N_{max}$. Thus, the number of alive sensors N_a is greater than N_{max}.

Next, we prove that if the guard inequality is satisfied at the beginning of a phase, a duty-cycle schedule can be found such that the guard inequality is satisfied at the beginning of the next phase. Suppose the guard inequality (1) is satisfied at the beginning of the phase. We select N_{max} sensors for the phase as follows: 1) $min(N_{max}, m)$ sensors with the least energy in list L_1, and 2) $max(N_{max} - m, 0)$ sensors with the least energy in list L_2. Then, at the beginning of the next phase, we construct a new List L', decide the new turning point m' and N'_a, and divide L' into three sublists L'_1, L'_2, and L'_3. Assuming the remaining energy in any sensor u_i $(0 \le i \le N_d - 1)$ in L' is e'_i, we consider two cases (Note that m' cannot be less than m.):

- $m' = m$. We have

$$\sum_{i=m'}^{N'_a-1} e'_i \ge (N_{max} - m')(t-1)\delta.$$

- $m' > m$. We have

$$\sum_{i=m'}^{N'_a-1} e'_i = \sum_{i=m}^{N'_a-1} e'_i - \sum_{i=m}^{m'-1} e'_i.$$

Clearly, for sensor $u_i (m \le i \le m' - 1)$, it holds that $e'_i = (t-1)\delta$. It follows

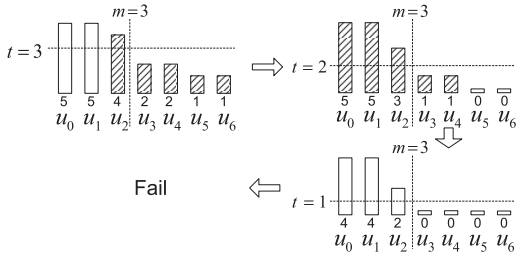

Fig. 2. A counterexample. Each bar represents a sensor, whose residual energy is marked by the value under it. Shaded bars represent sensors that are chosen to be active in the current phase. We assume $\delta = 1$.

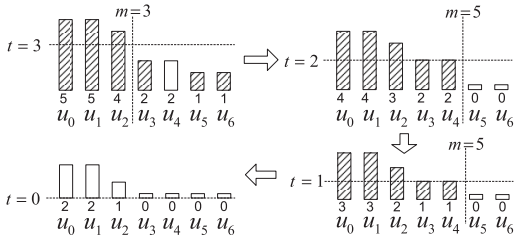

Fig. 3. Local-tier scheduling. Each bar represents a sensor, whose residual energy is marked by the value under it. Shaded bars are chosen to be active in the previous phase. We assume $\delta = 1$.

$$\begin{aligned}\sum_{i=m'}^{N'_a-1} e'_i &= \sum_{i=m}^{N'_a-1} e'_i - (m'-m)(t-1)\delta \\ &\geq (N_{max}-m)(t-1)\delta - (m'-m)(t-1)\delta \\ &= (N_{max}-m')(t-1)\delta.\end{aligned}$$

Thus, the guard inequality is satisfied at the beginning of the next phase. The correctness of Theorem 4.1 is straightforward due to the above two results. □

To further explain Theorem 4.1, a counterexample is shown in Fig. 2 to illustrate that, if the guard inequality is not satisfied in scheduling, the quality of service could be violated. Here, $N_{max} = 5$, the number of remaining phases before the MR's next visit (i.e., t) is 3, and for all the remaining phases, surveillance number is 5. At the beginning of the first phase, the guard inequality is satisfied. If a greedy scheduling policy is deployed, which always schedules sensors with the least residual energy, then at the beginning of the second phase, the left side of the guard inequality, i.e., the summation of u_3 and u_4's residual energy, is 2, and the right side of the guard inequality is 4. Thus, the guard inequality is violated at the beginning of the second phase, and at the beginning of third phase, there are only three alive sensors, but surveillance number is 5. As a result, the scheduling fails.

4.2 Controlled-Greedy Algorithm

Fig. 2 shows that the greedy scheduling policy cannot always satisfy the guard inequality at the beginning of any phase. In order to address this issue, we propose a controlled-greedy scheduling algorithm. The basic idea of the controlled-greedy scheduling algorithm is that if the greedy scheduling policy can satisfy the guard inequality, we apply this scheduling policy; otherwise, we schedule some sensors with higher remaining energy in order to satisfy the guard inequality.

Suppose phase p is t phases before the MR's next visit, the surveillance number for phase p is x, and the guard inequality is satisfied at the beginning of phase p. Our proposed controlled-greedy algorithm will schedule the duty cycles of sensors such that the guard inequality is still satisfied at the beginning of phase $p+1$. The details of the algorithm are as follows:

The first step is an attempted *Greedy Scheduling*. Among all the sensors alive, x sensors with the least energy are chosen. We simulate that these sensors are scheduled to be active for this phase. Hence, the energy of these sensors is deducted by δ. Then, we simulate that, at the beginning of phase $p+1$, based on the new energy level, lists L_1 and L_2 are constructed, and the guard inequality is tested. If the guard inequality is satisfied, meaning the attempt for greedy scheduling succeeds, the x chosen sensors are *really* scheduled to be active for phase p. Otherwise, the attempted greedy scheduling fails and we proceed to the second step.

The second step is *Semi-greedy Scheduling*. Note that when the semi-greedy scheduling is needed, we must have $m < N_{max}$ and $x > N_{max} - m$ at the beginning of phase p; otherwise, the greedy scheduling will succeed. The semi-greedy scheduling schedules the $x - (N_{max} - m)$ sensors with the least energy in L_1, and $N_{max} - m$ sensors with the least energy in L_2.

We use an example in Fig. 3 to illustrate the algorithm. As shown in Fig. 3, at the beginning of the first phase, we have $t = 3, m = 3, N_{max} = 5$, and for all the remaining phases, surveillance number is 5, i.e., $x = 5$. The first step fails, so we proceed to the second step. Since $x = 5 > (N_{max} - m) = 2$, we schedule $x - (N_{max} - m) = 3$ least energy supplied sensors in L_1, i.e., u_0, u_1, and u_2. The other two sensors are the two sensors with the least energy in L_2, i.e., u_5 and u_6. For the second and third phases, greedy scheduling succeeds, and the quality of service is satisfied until the MR's next visit. Algorithm 1 formally presents the above local-tier scheduling algorithm.

Algorithm 1. Local-Tier Scheduling in a group at phase p

Notations:

x: surveillance number in phase p

t: the number of remaining phases before the MR's next visit

L, L_1, L_2, L_3, L': lists of sensors in an energy decreasing sequence

e_i, e'_i: the amount of the remaining energy of the i^{th} sensor in list L, L' (respectively)

δ, N_d: defined previously

1: Calculate m and partition L into three sublists: L_1, L_2, L_3. $|L_1| = m, \forall e_i \in L_1, e_i \geq t\delta$; $\forall e_i \in L_2$, $\delta \leq e_i < t\delta$.

2: Tentatively schedule the last x alive sensors in L to be active, and decrement their energy by δ.

3: Construct a new list L' and decide the new turning point m' and N'_a.

4: **if** $\sum_{i=m'}^{N'_a-1} e'_i \geq (N_{max} - m')(t-1)\delta$ **then**

5: Confirm the schedule and return.

6: **else**

7: Cancel the changes made in line 2.

8: Schedule the last $(x-(N_{max}-m))$ sensors in L_1.
9: Schedule the last $(N_{max}-m)$ sensors in L_2.

In Algorithm 1, Lines (1), (3), and (4) each take $O(N_d)$ time, and other lines take less time. Therefore, the time complexity of Algorithm 1 is $O(N_d)$.

4.3 Correctness of the Algorithm

Theorem 4.2. *Our local-tier scheduling algorithm ensures that if the guard inequality is satisfied at the beginning of the first phase right after an replacement/reclamation, the guard inequality is satisfied at the beginning of every phase before the next replacement/reclamation.*

Proof. We prove this claim by induction. We are already given that the guard inequality is satisfied at the beginning of the first phase.

Assume the guard inequality is satisfied at the beginning of phase p, we show that after applying our local-tier scheduling algorithm, the guard inequality is satisfied at the beginning of phase $p+1$. We consider two cases:

- *Greedy scheduling succeeds*. If the greedy scheduling succeeds, the test at Line (4) in Algorithm 1 must return true. Thus, the guard inequality is satisfied at the beginning of the next phase.
- *Greedy scheduling fails*. If the greedy scheduling fails, the semi-greedy scheduling will be applied, i.e., we schedule the last $(x-(N_{max}-m))$ sensors in L_1, and the last $(N_{max}-m)$ sensors in L_2. Then, at the beginning of the next phase, we construct a new List L' and decide the new turning point m', which divides sublists L'_1 and L'_2. We consider two cases: 1) $m=m'$ and 2) $m'>m$. With the similar derivation in the proof of Theorem 4.1, we conclude that the guard inequality is satisfied at the beginning of the next phase.

Therefore, our local-tier scheduling algorithm guarantees that the guard inequality is satisfied for every phase between two consecutive replacements if the guard inequality is satisfied at the beginning of the first phase. □

At the beginning of the sensor network's life, every node has full energy and the guard inequality is satisfied. After that, our global-tier scheduling algorithm guarantees that, after each reclamation/replacement, the inequality is also satisfied. Therefore, during the whole life of the sensor network, the inequality is satisfied and the quality of surveillance service is satisfied.

5 Global-Tier Scheduling

At the end of each round j, after the MR completes reclamation, it knows the amount of residual energy of each group, i.e., the summation of residual energy on all sensors in the group. Based on the amount, the MR first calculates the number of sensors to be reclaimed/replaced for the group in round $j+2$, which is called the *replacement number*. Based on the replacement numbers and the residual energy information of all groups, the MR runs our proposed MR travel scheduling algorithm to calculate a schedule to visit every group in round $j+2$. When the MR visits a group in the coming round (round $j+1$), it notifies the group of its visiting time and replacement number in round $j+2$. Meanwhile, the MR collects the residual energy information from each sensor in the group for the global-tier scheduling for round $j+3$. Assuming that the MR and each sensor in a group can reach each other in one hop, the communication complexity between the MR and a group is N_d+1 messages in one round.

With the prerequisite of satisfying the quality of service, the global-tier scheduling has two objectives: 1) the total travel distance of the MR is minimized, and 2) the remaining energy of sensors to be replaced is minimized. Satisfying the above objectives makes the MR travel scheduling an NP-hard problem. In the following, we present how to calculate the replacement number, prove the NP-hard nature of calculating the MR's travel schedule, and present heuristic solutions to this problem.

5.1 Calculation of Replacement Numbers

The replacement number at group $g_i (1 \le i \le n)$ in round $j (j \ge 1)$ is denoted as $N_r(i,j)$. When the MR comes to group g_i, it chooses $N_r(i,j)$ sensors with the minimum residual energy to reclaim, and replace them with the same number of fully charged sensors. Let $E(i,j)$ denote the summation of residual energy on all sensors in group g_i after the MR completes replacement/reclamation at group g_i in round j and e be the amount of energy held by a fully charged sensor. Recall that, l is the length of a round and τ is the lifetime of a sensor if it is active all the time. $N_r(i,j)$ is conservatively calculated as follows:

$$\begin{cases} N_r(i,1) = N_r(i,2) = max\{\lceil (l/\tau)N_{max}\rceil, N_{max}\} \\ N_r(i,j+2) = \\ \quad max\Big\{\lceil 3(l/\tau)N_{max}\rceil - N_r(i,j+1) - \Big\lfloor \frac{E(i,j)}{e}\Big\rfloor, 0\Big\} \quad (j \ge 1). \end{cases}$$

For the first two rounds, the replacement number is predetermined as $max\{\lceil (l/\tau)N_{max}\rceil, N_{max}\}$ to guard against the worst case scenario when N_{max} nodes are needed to be active all the time. The replacement number of round 3 and later is calculated by the second formula. The part $\lceil 3(l/\tau)N_{max}\rceil - N_r(i,j+1)$ reflects that in the worst case, group g_i may be required to have the maximum number of active sensors for three rounds in a row, rounds $j, j+1$, and $j+2$, but with only one replenishment in round $j+1$. This happens when the MR only visits group g_i at the beginning of round j and at the end of round $j+2$. $\lfloor \frac{E(i,j)}{e}\rfloor$ is the number of fully charged sensors that have equivalent energy to the total residual energy in group g_i. Note that, if the MR does not visit group g_i in round j, it cannot obtain the exact number of $E(i,j)$ and will conservatively assume that N_{max} sensors are required to be active throughout round j. If $\lceil 3(l/\tau)N_{max}\rceil - N_r(i,j+1) - \lfloor \frac{E(i,j)}{e}\rfloor < 0$, then $N_r(i,j+2)=0$, which implies that group g_i does not need replacement/reclamation in round $j+2$ and the MR does not visit group g_i in the round.

5.2 Calculation of the Travel Schedule for the MR

For each round, we need to 1) minimize the total travel distance of the MR in this round and 2) minimize the residual

group \ phase	1	2	...	m
g_1	$e(1,1)$	$e(1,2)$	... 0 ... 0 / ...	/
g_2	$e(2,1)$	$e(2,2)$	... 0 ... 0 / ...	/
⋮	⋮	⋮	⋮	⋮
g_x	$e(x,1)$	$e(x,2)$	...	$e(x,m)$
⋮	⋮	⋮	⋮	⋮
g_n	$e(n,1)$	$e(n,2)$	... 0 ... 0 / ...	/

Fig. 4. Table R: Residual energy in sensors to be reclaimed/replaced in all groups at all phases in round $j+2$. "/" represents invalid number.

energy in sensors to be reclaimed/replaced. The reason for objective 2 is that if we minimize the residual energy in sensors to be reclaimed/replaced, the net amount of energy replenished into the network in this round is maximized, and thus the demand for future reclamation/replacement is minimized. As a result, the MR will need to replace the smallest number of sensors in future rounds, and thus the total MR travel distance is reduced in the long run.

Since minimizing the residual energy of sensors to be reclaimed/replaced affects the travel distance of the MR, and vice versa, these two objectives are not likely to be satisfied at the same time. Hence, we associate a weight with each objective and aim to optimize the combined objective. In the following, we introduce related data structures, formally state this optimization problem, prove its NP-completeness, and present heuristic solutions.

5.2.1 Data Structures

At the end of round j, the MR calculates its visiting schedule to all the groups in round $j+2$. A 2D table R, is used in calculating the best visiting time of the MR to each group. R records the total residual energy in the sensors to be reclaimed/replaced in each group at each phase of round $j+2$ if the MR visits the group at the phase. The energy can be positive, zero, or an invalid number. The invalid number means the quality of service is already violated in the phase because the MR comes too late. Fig. 4 shows an example. Each entry $e(i,p)$ in the table represents the total residual energy in the sensors to be reclaimed/replaced in group g_i when the MR visits it at phase p in round $j+2$. Along a row, the value of $e(i,p)$ gets lower as p increases, i.e., $e(i,1) > e(i,2) > \cdots > e(i,c_i) = \cdots = e(i,d_i) = 0$. The best phases for the MR to visit group i in round $j+2$ are within $[c_i, d_i]$, for which the residual energy is 0. Phase d_i is called the *deadline* for the MR to visit group g_i in round $j+2$, because if the MR comes late, the quality of service is violated. It is also possible that even if the MR visits a group at the last phase of round $j+2$, the residual energy is still greater than 0. Group g_x is such an example. In the case, we call the last phase, i.e., phase m, the deadline for the MR to visit group g_x in round $j+2$, i.e., $d_x = m$.

Because the future energy consumption pattern is not known beforehand, the table can only be constructed based on prediction. In our scheme, we simulate the local-tier scheduling algorithm phase by phase until phase p in round $j+2$, using the maximum surveillance numbers (N_{max}) conservatively to obtain entry $e(i,p)$. Other methods to predict surveillance numbers will be explored in our future work.

In addition to the 2D table R, the following data structures are needed in formalizing the problem:

- $G(V, E, W(V), W(E), R)$ denotes a complete undirected graph. $V = \{g_i \mid 0 \le i \le n\}$, where g_0 represents the ES and $g_1, g_2, \ldots, g_n$ represent n groups. $W(V) = \{N_r(1, j+2), \ldots, N_r(n, j+2)\}$, where $N_r(i, j+2)$ is the replacement number for group g_i $(1 \le i \le n)$ in round $j+2$. For any two groups g_i and $g_k (0 \le i \ne k \le n)$, there is an edge (g_i, g_k), whose weight represents the cost for the MR's travel between groups g_i and g_k, and $W(E)$ stores the cost of each such edge. R is defined above.
- $\vec{t} = (t_1, t_2, \ldots, t_n)$ is a visiting time vector in which t_i is the phase when the MR visits group g_i in round $j+2$.
- $D = d(\vec{t})$ is the total traveling distance for the MR to fulfill a complete reclamation/replacement tour according to time vector $\vec{t}$. Note that the MR needs to go back to the ES for reloading multiple times due to the limited capacity of the MR.

5.2.2 Nature of MR Travel Scheduling Problem

The problem is to find a time vector $\vec{t}$ for the MR which can carry up to C fully charged sensors to visit the groups in the network and replace $W(V)$ sensors, such that the following metric is minimized.

$$Y = \alpha D + \beta \sum_{i=1}^{n} e(i, t_i) = \alpha d(\vec{t}) + \beta \sum_{i=1}^{n} e(i, t_i), \qquad (2)$$

where α and β are two system parameters, each representing the weight of the associated item.

The MR travel scheduling problem is an NP-hard problem since the well-known NP-hard Vehicle Routing Problem with Time Windows (VRPTW) [9], [10], [11] can be reduced to this problem. The proof can be found in [12].

5.2.3 Heuristic Solutions

The object value Y depends on two optimization objectives, the total MR travel distance and the total residual energy in sensors to be reclaimed/replaced, which cannot be optimized at the same time. Hence, we propose heuristic solutions to optimize the two objectives according to their associated weights, α and β.

If α is much larger than β, distance D is minimized with priority. As the capacity of the MR is limited, multiple tours to visit all the groups are calculated to minimize D, where each tour is a sequence of groups that the MR should visit in order. Since changing the ordering of different tours does not affect D, we can find an order of these tours to minimize the residual energy in sensors to be reclaimed/replaced based on table R.

If β is much larger than α, the residual energy in sensors to be reclaimed/replaced is minimized with priority. Intuitively, groups with similar deadlines should be visited at similar time. This can be done by distributing groups into tours based on the similarity of their deadlines. The MR will conduct the tours in the increasing order of the deadlines; i.e., tours with early deadlines will be conducted early. Within each tour, a visiting order of the groups can be determined to minimize the MR travel distance.

For more general cases, we propose a *supertour heuristic solution* and use a parameter M to reflect the relative weight of α and β. The basic idea of the supertour heuristic solution

is as follows: given n groups, we distribute them into a number of *supertours* based on similarity of their deadlines. A supertour is composed of M *physical* tours. That is, in a supertour, the MR needs to go back to the ES for $M-1$ times for reloading. We order supertours according to the deadlines aiming to minimize the total amount of residual energy in sensors to be reclaimed/replaced. Inside each supertour, we first calculate multiple physical tours to minimize the total travel distance; then we determine the sequence of these physical tours according to the deadlines to minimize the residual energy.

When M gets larger, the total travel distance tends to be smaller, but the total amount of residual energy in sensors to be reclaimed/replaced tends to be larger, and vice versa. Note that when $M=1$, the solution is reduced to the case that the residual energy is minimized first. M is tuned according to the weights α and β such that Y is minimized.

Algorithm 2 formally presents the supertour heuristic solution. Step 4 in Algorithm 2 is to divide a supertour into M physical tours such that the travel distance is minimized, which is the Capacitated Vehicle Routing Problem (CVRP) problem, an NP-hard problem. We employ a well-known and effective heuristic algorithm proposed in [13] to solve it.

Algorithm 2. Super-Tour Heuristic Solution (for round $j+2$)

Notations:

M: the number of physical tours in a super tour
C: the capacity of the MR
d_i: the deadline of group i in round $j+2$
$e(i,d_i)$: the amount of residual energy of group i if it is visited by the MR at phase $d(i)$ in round $j+2$

1: Sort all groups g_i into a sequence L in an increasing order based on pair $\langle d_i, e(i,d_i)\rangle$ $(1 \le i \le n)$.
2: **while** L is *NOT* empty **do**
3: Construct a super tour ST by reversely traversing L such that the sum of their demands is less than or equal to $M * C$ while adding one more group will make the sum greater than $M * C$.
4: Divide ST into M physical tours such that the overall travel distance is minimized.
5: Decide the visiting times to all groups in ST.
6: Remove all groups in ST from L.

In Algorithm 2, Line (1) takes $O(n)$ time if the counting sort algorithm is adopted. Lines (3) and (6) take $O(n)$ time as well. Lines (4) and (5) are executed for at most n times. If the time to run an instance of CVRP with n stations is $T(n)$, the complexity for running lines (4) and (5) is $O(nT(n))$. Therefore, the overall complexity of the algorithm is $O(nT(n))$. With the heuristic CVRP solver [13] that we use, $T(n)$ is an exponential function of n in the worst case but it is a polynomial function in most cases.

5.2.4 Two Special Topologies

It is an NP-hard problem to deal with arbitrary topology and divide a supertour into M physical tours in step 4 in Algorithm 2, and generally some complex heuristics are necessary. However, we notice that there are two deployments used in a number of applications of sensor networks which can be handled in a simpler way. One is the line topology. The examples include sensor networks deployed along a highway [14] to monitor the traffic or deployed along a boundary [15] to detect intrusion. Another is the ring topology. For example, sensor networks deployed around a lake [16] exhibit such a topology. In this section, we introduce two efficient algorithms to handle these specific scenarios optimally.

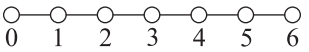

Fig. 5. Line example.

Line topology. Fig. 5 shows an example of line topology. Suppose posts $1, 2, \ldots, 6$ are located on a line from left to right. The ES is to the left of post 1. When the MR makes a physical tour, it is efficient for it to travel a forward path and return in the reverse path without zigzags. In this case, the travel distance of a physical tour is determined by the furthest group from the ES.

Suppose a supertour includes M physical tours. We define FG_{t_i} as the group which is farthest away from the ES among the groups to be visited in physical tour i $(i=1,2,\ldots,M)$ and define $d(FG_{t_i})$ as its distance to the ES. We assume all tours t_i $(i=1,\ldots,M)$ are sorted according to the decreasing order of $d(FG_{t_i})$, i.e., $d(FG_{t_M})$ has the smallest value, while $d(FG_{t_1})$ has the largest value.

Given M tours, we aim to determine $FG_{t_i}, i=1, 2, \ldots, M$, such that

$$\sum_{i=1}^{M} d(FG_{t_i})$$

is minimized.

Our solution to this problem is as follows: starting from group FG_{t_1}, which is the farthest group from the ES, we examine groups one by one along the direction toward the ES, summing up the demands of the examined groups until the sum reaches C. These groups form a physical tour. Among the remaining groups that have not been fully replenished by tour t_1, the group furthest away from the ES becomes FG_{t_2}. Then, we start from FG_{t_2}, and count the second physical tour in the same way as determining the first tour. As this process continues, tours $t_2, \ldots, t_M$ are formed one by one, until the ES is reached. Based on this idea, a greedy algorithm can be developed to make the optimal travel schedule.

Algorithm 3 formally presents how to determine FG_{t_i}, where $i=1,2,\ldots,M$. The input for Algorithm 3 is a supertour $(g_1, g_2, \ldots, g_s)$ which is formed in step 3 of Algorithm 2, where g_1 is the nearest group to the ES, and g_s is the farthest group to the ES.

Algorithm 3. Algorithm for Line (4) in Algorithm 2 for Line Topology

Notations:

$FG_{t_i}, d(FG_{t_i})$: defined before
PT_i: groups to be visited in physical tour $t_i, 1 \le i \le M$
g_k: the k^{th} group in the super tour, indexed by the distance to the ES in an increasing order, $1 \le k \le s$
$N_r(k)$: the replacement number of group g_k
D: total MR travel distance

1: $FG_{t_1} \leftarrow g_s, sum \leftarrow 0, start \leftarrow FG_{t_1}, D \leftarrow 0$

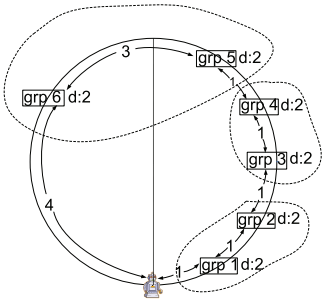

Fig. 6. Ring example. A rectangle represents a group. The number by the side of each group is its demand. The number on an arc is the distance between two adjacent groups. We assume $C = 4$.

2: $i \leftarrow 1, k \leftarrow s$
3: **while** $k \geq 1$ **do**
4: $sum \leftarrow sum + N_r(k)$
5: **if** $sum > C$ **then**
6: **if** $(sum - N_r(k)) = C$ **then**
7: $PT_i = \{g_{start}, \ldots, g_{k+1}\}$
8: **else**
9: /* group g_k will be visited in both tour t_i and t_{i+1} */
10: /* generate a new group $g_{k'}$ */
11: $N_r(k') \leftarrow C - (sum - N_r(k))$
12: $PT_i = \{g_{start}, \ldots, g_{k+1}\} \cup g_{k'}$
13: $N_r(k) = sum - C$
14: $sum \leftarrow 0, start \leftarrow k, i \leftarrow i + 1, D \leftarrow D + d(FG_{t_i})*2$
15: **else**
16: $k \leftarrow k - 1$
17: $PT_i = \{g_{start}, \ldots, g_{k+1}\}, D \leftarrow D + d(FG_{t_i})*2$
18: RETURN $(\{t_1, t_2, \ldots, t_M\}, D)$

In Algorithm 3, the *while* loop from Line (3) to Line (16) dominates the running time, and each statement in the loop is executed at most once. Therefore, the time complexity of Algorithm 3 is $O(s)$. As a result, if we use Algorithm 3 to implement Step 4 in Algorithm 2 for the line topology, the time complexity is $O(n + \sum_{i=0}^{r} s_i) = O(n)$, where r is the total number of supertours in Algorithm 2 and s_i is the number of groups in supertour i.

Ring topology. Fig. 6 shows an example of ring topology. For the ring topology, we still employ a greedy heuristic. Our algorithm is described as follows: given each group in a supertour, we consider it as a start point, sum up the demands of groups until the sum reaches C in the counterclockwise direction to form the first physical tour, and then continue to form the second physical tour, and so on and so forth. After all the physical tours are formed, we compute the total travel distance. We will choose the set of physical tours which has the minimum total travel distance. Algorithm 4 shows this process.

Algorithm 4. Algorithm for Line (4) in Algorithm 2 for Ring Topology

Notations:

S: current set of physical tours
S_{min}: the set of physical tours with the minimum travel distance
D': total MR travel distance of S
D'_{min}: total MR travel distance of S_{min}
g_k: the k^{th} group in the super tour, $1 \leq k \leq s$. g_k and g_{k+1} are physically adjacent

1: $S_{min} = \phi, D'_{min} = \infty$
2: **for** $k \leftarrow 1$ to s **do**
3: reorder $(g_1, g_2, \ldots, g_s)$ as $(g_k, \ldots, g_s, g_1, \ldots, g_{k-1})$
4: call a modified version of Algorithm 3 with sequence $(g_k, \ldots, g_s, g_1, \ldots, g_{k-1})$
5: $(S, D') = (\{t_1, t_2, \ldots, t_M\}, D)$
6: **if** $D' < D'_{min}$ **then**
7: $S_{min} = S, D'_{min} = D'$
8: RETURN (S_{min}, D'_{min})

Line (4) in Algorithm 4 calls a modified version of Algorithm 3. The differences between this modified version and the original version are as follows:

- In the modified version, for any tour, if all groups are on one half of the ring, the MR's travel distance in this tour is twice the distance of the farthest group to the ES.
- In the modified version, if some groups in a tour are on the left half ring, and other groups are on the right half ring, we consider two cases. Assume that on the left half ring, the farthest group from the ES is g_l with distance $d(g_l)$, and on the right half ring, the farthest group from the ES is g_r with distance $d(g_r)$.
  - If the distance between g_l and g_r across the boundary opposite to the ES of the two half rings is less than half of the perimeter of the ring, the tour becomes the whole ring, and the MR travel distance is the perimeter of the ring.
  - Otherwise, the tour is composed of two subtours, each of which starts from the ES, goes to either g_l or g_r, and goes back to the ES. The MR travel distance is $2(d(g_l) + d(g_r))$.

The time complexity of Algorithm 4 is $O(s^2)$ since Lines (2) to (7) run for $O(s)$ times, and each execution of Line (4) takes $O(s)$. When we use Algorithm 4 to implement Step 4 in Algorithm 2, the time complexity of Algorithm 2 becomes $O(n + \sum_{i=0}^{r} s_i^2) \leq O(n^2)$, recalling r is the total number of supertours and s_i is the number of groups in supertour i.

In the following, we show that Algorithm 4 in fact returns the optimal set of physical tours for ring topology.

Theorem 5.1. *Algorithm 4 returns a visiting scheme with travel distance the same as theoretical optimum value for ring topology.*

Proof. Let S be the physical tour set returned by the greedy algorithm, and its travel distance is D_S. Now given any other physical tour set S', which is not returned by the greedy algorithm, we claim that $D_{S'} \geq D_S$.

By examining the property of the greedy algorithm, it explores all possible consecutive neighboring groups into physical tours, and returns a physical tour set with the minimum travel distance.

Hence, in set S', there must be some physical tours, in which groups are not consecutive. For instance, in Fig. 7, physical tour t_i contains three groups: A, B, and E,

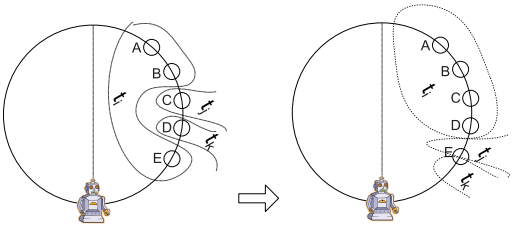

Fig. 7. Example in proof of Theorem 5.1.

which are not consecutive. However, we can transform the nonconsecutive physical tours into consecutive ones without increasing the travel distance of S'. The basic idea is as follows: we first pick the group that is farthest from the ES, and assume this group is in tour t_i in S'. We next exchange nonconsecutive groups in t_i for consecutive ones in other groups. Take Fig. 7 as an example. Suppose A is the farthest group from the ES, and $N_r(C), N_r(D)$, and $N_r(E)$ represent the replacement numbers of group C, D, and E, respectively, with equation $N_r(E) = N_r(C) + N_r(D)$ satisfied. Assuming group C is in tour t_j, and group D in group t_k, t_i will exchange $N_r(C)$ sensors of group E for group C in t_j, and $N_r(D)$ sensors of group E for group D in t_k. Note that group E is in both tours t_j and t_k after the transformation. As a result, none of the physical tours in S' has its travel distance increased.

Repeatedly applying the above steps to other nonconsecutive groups, we will finally get a set of physical tours which only contain consecutive neighboring groups without increasing the travel distance. Hence, the greedy algorithm returns a solution which is no worse than the optimum value. □

6 Performance Evaluation

6.1 Experimental Methodology

We built a custom simulator using C++ to evaluate the performance of the proposed scheme. In all experiments, we normalize the full energy level of a sensor to 400 units and the energy consumption rate to 0.1 unit/minute. Thus, each sensor's lifetime τ is 4,000 minutes. The length of a phase is set to 10 minutes. The length of a round l is set to 4,000 minutes except the experiment studying the impact of the number of sensors deployed in a group, i.e., group size, on the performance. We set $N_{min} = 6$ and $N_{max} = 16$ for all groups except the comparative study between our scheme and the optimum solution on a small-scale network.

Except the experiment studying the impact of group size on the performance, group size is set to 32. Note that when N_{max} is 16 and a sensor's lifetime is equal to a round length, a group of 32 sensors means initially deployed sensors are able to guarantee the quality of service for two rounds in the worst case scenario. The MR has a capacity of 80 sensors, and its speed is 20 meters/minute unless otherwise mentioned.

In practice, surveillance number is determined by the application, as well as the real-time frequency and distribution of events. In our simulation, we consider two types of distributions of the number: the linear decrease distribution and the Gaussian distribution.

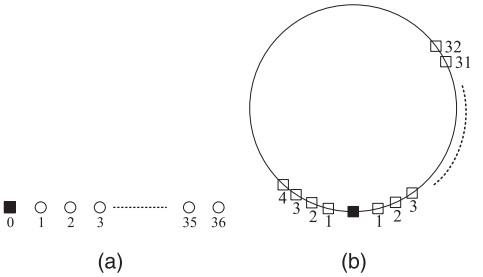

Fig. 8. Line and ring topology for experimental setting. The solid square is the ES, and hollow squares are groups. The number by each group is its distance to the ES. (a) Line. (b) Ring.

- *Linear decrease*. The Probability Density Function *(pdf)* of surveillance number linearly decreases as surveillance number increases. Specifically, the *pdf* for surveillance number $i \in [N_{min}, N_{max}]$ is

$$f(i) = \frac{N_{max} - i + 1}{\sum_{k=N_{min}}^{N_{max}} (k - N_{min} + 1)}.$$

- *Gaussian*. We deploy Gaussian distribution $N(\mu = N_{min}, \sigma^2 = 4)$ and truncate it to the range $[N_{min}, N_{max}]$.

We conducted the following sets of experiments: studying the trade-off between the residual energy in sensors to be reclaimed/replaced and the MR's travel distance, comparing the supertour heuristic solution and a naive solution, comparing the supertour heuristic solution and the optimal solution, studying the impact of MR's capacity on the performance, and studying the impact of group size on the performance. The first set of experiments is conducted on three network topologies: line, ring, and arbitrary, while other experiments are conducted on the arbitrary topology. For each topology, 36 groups of sensors are distributed in the network field. For each experiment, our algorithm is executed for a long time period, starting at 0 and ending at a *cutoff* time. The cutoff time is set to 40,000 minutes for all experiments. Furthermore, we run each simulation for 20 times and take the 95 percent confidence interval for each evaluated metric.

6.2 Simulation Results

6.2.1 Trade-Off between Residual Energy in Sensors and MR's Travel Distance

In this section, we study the trade-off between the amount of residual energy in sensors to be reclaimed/replaced and the MR's travel distance, by measuring and showing the energy amount and the travel distance as functions of the number of physical tours in a supertour (i.e., system parameter M).

Line topology. We deploy 36 sensor groups with even interval on a straight line with a length of $3{,}600$ m as shown in Fig. 8a. The energy station is located at the position 0. Figs. 9a and 9b show how the total residual energy in reclaimed/replaced sensors and MR's travel distance changes as M varies. As can be seen, when M increases, the total residual energy in sensors to be reclaimed/replaced increases, while the total travel distance decreases. The value of M leverages the two optimization factors in (2). When $M = 1$, it corresponds to the energy-first heuristic, since there is only

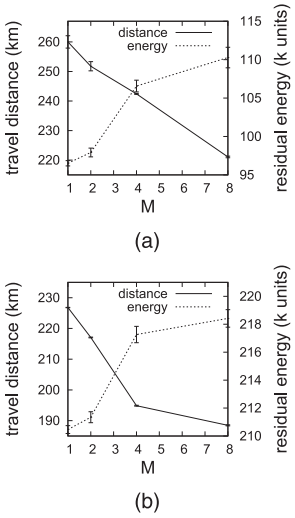

Fig. 9. Impact of number of physical tours in a supertour. (a) Line: linear decrease. (b) Line: Gaussian.

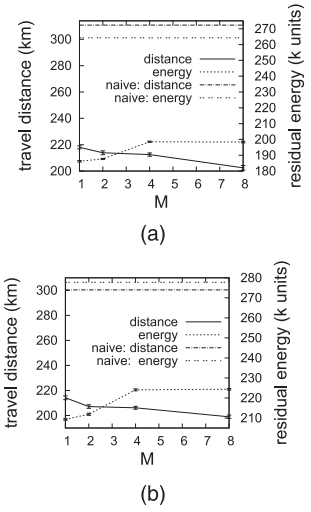

Fig. 11. Impact of M. (a) Arbitrary: linear decrease. (b) Arbitrary: Gaussian.

one physical tour in a supertour; while $M = 8$, it corresponds to the distance-first heuristic since the MR can visit all the sensors in one supertour.

Ring topology. We deploy 36 sensor groups along a circle with a perimeter 7,200 m as shown in Fig. 8b. Thirty-two sensor groups are on the right side of the circle, and four sensors are on the left side of the circle. Any two adjacent groups have a distance of 100 m. Figs. 10a and 10b show how the residual energy in sensors to be reclaimed/replaced and the MR's travel distance changes as the number of physical tours in a supertour varies. The figures illustrate the same trend as in the line topology case.

Arbitrary topology. In this experiment, we study in the arbitrary topology the trade-off between the amount of residual energy in sensors to be reclaimed/replaced and the MR's travel distance, as M varies. At the same time, we also compare our supertour heuristic solution with a naive solution. The naive solution works in the following way: the MR visits all groups in a fixed order in all rounds. In a round, whenever the MR has deployed all the fully charged sensors it carries, it goes back to the ES for reloading. Note that sensors in all groups still perform the proposed local-tier scheduling algorithm in this case.

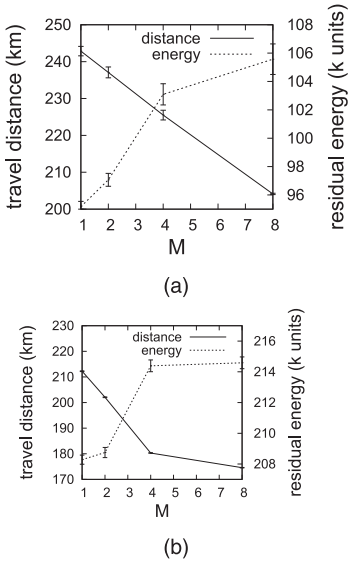

Fig. 10. Impact of number of physical tours in a supertour. (a) Ring: linear decrease. (b) Ring: Gaussian.

We deploy 36 groups of sensors in a $1{,}000\ \mathrm{m} * 1{,}000\ \mathrm{m}$ square field at random. Figs. 11a and 11b show how the amount of residual energy in sensors to be reclaimed/replaced and the MR's travel distance changes as M varies. As can be seen, the figures illustrate the same trend as in the line and ring topologies case. Furthermore, for all values of M, the both metrics of the supertour heuristic solution are considerably better than those of the naive solution. For instance, when $M = 2$ in the Gaussian distribution case, the travel distance and the amount of residual energy in sensors to be reclaimed/replaced of the supertour heuristic solution are 31 and 23.7 percent lower than those of the naive solution, respectively.

6.2.2 Comparison between Supertour Heuristic and Optimal Solution

Due to the NP-Completeness nature of the problem, we cannot work out the optimal solution for a network with a practical size. However, for a network small enough, we can work out the optimal solution by enumerating all the possible travel schedules and get the best one. In the experiment, we randomly deploy nine groups into a $500\ \mathrm{m} * 500\ \mathrm{m}$ square field. The MR has a capacity of 10 sensors, and its speed is set to 10 meters per minute. In the experiment, we use object value Y in (2) with $\alpha = 5$ and $\beta = 1$ as the performance metric.

Fig. 12 shows the performance difference between the supertour heuristic solution and the optimal solution in the Gaussian distribution case. The y-axis of Fig. 12 is object value Y in (2). From the figure, we can see, when $M = 4$, the object value of the supertour heuristic solution is only 0.8 percent higher than the optimal value. Even though the result is from a small size problem, we still can see the effectiveness of our proposed heuristic. Note that even though the objective value

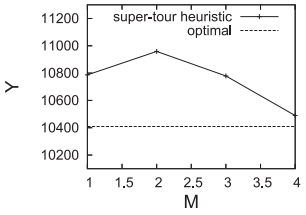

Fig. 12. Comparison between supertour heuristic and optimal solution.

is worse when M is smaller than 4, through simulation, the M which makes the best result can be found out and used.

6.2.3 Impact of MR's Capacity

In this experiment, we randomly deploy 36 groups of sensors in a $1{,}000\ \mathrm{m} * 1{,}000\ \mathrm{m}$ square field, and set $M = 2$. The capacity of the MR is varied among $\{40, 50, 60, 70, 80\}$ sensors. As can be seen from Fig. 13, both performance metrics decrease as the MR carries more sensors. With higher capacity, the MR can pay less number of trips to finish the replenishment process, and thus travel shorter in total. On the other hand, since the MR is able to finish the replenishment process in a shorter time, groups tend to be visited later by the MR and thus have more chance to use the energy in the sensors to be reclaimed/replaced. In general, larger capacity is beneficial.

6.2.4 Impact of Group Size

In this experiment, we also deploy 36 groups of sensors in a $1{,}000\ \mathrm{m} * 1{,}000\ \mathrm{m}$ square field at random, and set $M = 2$. Group size is varied among $\{24, 28, 32, 36, 40\}$ sensors. Round length l also varies with group size since larger group size can allow more delay for replacement given fixed N_{max}. Given a group size s, s and l satisfy $s = \lceil 2(l/\tau)N_{max} \rceil$.

In Fig. 14, both the total residual energy and the total travel distance decrease as round length increases. The decrease of the former metric is because longer round length (as a result of larger group size) gives groups more chances to use up the energy in the sensors before they are reclaimed/replaced. The decrease in total travel distance can be explained as follows: let us first consider the average length of a tour. When round length increases,

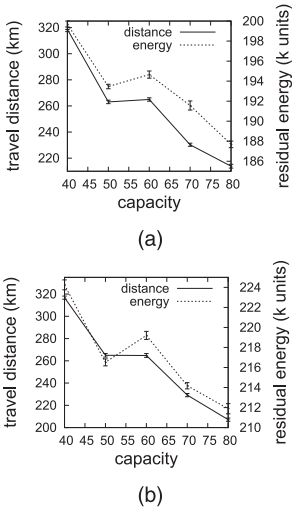

Fig. 13. Impact of capacity. (a) Linear decrease. (b) Gaussian.

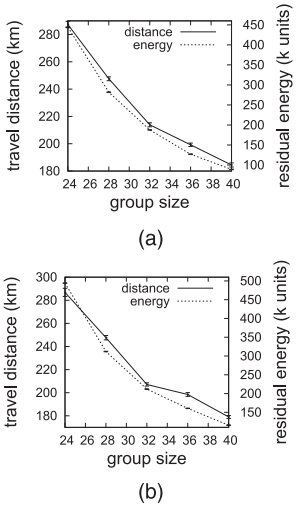

Fig. 14. Impact of group size. (a) Linear decrease. (b) Gaussian.

each group has more sensors to be replaced in one round, which implies a less number of visiting destinations in each tour and hence shorter distance for each tour. Next, consider the number of tours during the entire simulation time (40,000 minutes). As round length increases, the number of tours needed in a round increases; but the total number of rounds decreases given the fixed cutoff time. The total number of tours is the result of joint effects of these two conflicting factors. By checking simulation traces, we find that the number of tours generally decreases as round length increases. Since both the average length of a tour and the total number of tours decrease, the total travel distance decreases.

This experiment shows that larger group size reduces the total MR travel distance in the long run.

7 Discussions

We now discuss some practical issues in implementing the ARTS scheme. First, the ARTS scheme treats failed sensors the same as sensors drained of energy, i.e., failed sensors will be replaced by the MR. We employ the following method to detect sensor failures. At the time for scheduling (i.e., at the beginning of a phase), a sensor u that is chosen to be active in the phase, should broadcast a message to all sensors in the group. Other sensors also know which sensors shall be active for the phase. If they do not receive the message from sensor u, they assume sensor u has failed. Hence, they rerun the local-tier scheduling algorithm to select another active sensor to replace sensor u.

In the local-tier scheduling, sensors in a group may have different views regarding the amount of remaining energy in all sensors due to sensor failure or other reasons. To address this issue, we let each sensor broadcast its amount of remaining energy at the beginning of a phase every certain number of phases. When other sensors receive this information, they update their record accordingly.

8 Related Work

To prolong sensor network lifetime, duty-cycling schemes [6], [17], [18] have been proposed, with which sensor nodes

spend most of their lifetime in the sleeping state. They may slow down energy consumption in sensor nodes, but cannot replenish energy to the network. Therefore, the lifetime of the network is inherently limited by the amount of energy preloaded to sensor nodes. Schemes [1], [19], [20] have also been proposed to balance energy consumption among sensor nodes. However, they do not always pick the most energy-efficient sensor nodes to perform a certain task, and thus may consume more than necessary energy.

Researchers have studied environmental energy such as sunlight and acoustic vibrations [1], [2], [3], [4], [5], [21], [22], [23]. In these schemes, part or all sensor nodes are equipped with environmental energy harvesting devices, e.g., solar cells, and they harvest environmental energy in an opportunistic fashion. These schemes have the following practical issues: 1) They deeply rely on uncontrollable environment conditions. For instance, cloudy skies may prevent a sensor node from harvesting solar energy. 2) In some cases, the amount of environmental energy a sensor node can harvest is proportional to the size of the energy harvesting device. For instance, the energy that a solar cell can harvest is proportional to its surface area. It may be infeasible to equip a tiny sensor with large energy harvesting devices.

Numerous schemes [24], [25], [26] deploy sensor nodes incrementally to extend network lifetime. When sensor nodes fail or are drained of energy, a controller will deploy new sensor nodes to replace them. This approach seems to be a convenient solution; however, it is not environmental friendly or practical in many scenarios. For example, in the applications of natural environmental monitoring [27], continually deploying sensor nodes without reclaiming the deserted ones may pollute the environment.

Mobility-based data collection schemes are closely related to our scheme. In [28], the authors classified mobility-based data collection schemes into three categories: *the mobile base station-based schemes*, *the mobile data collector-based schemes*, and *the rendezvous-based schemes*. In the mobile base station-based approaches [19], [20], mobile base stations (i.e., sinks) need to move to different locations periodically to balance the network traffic. The frequent movement of base stations may consume large amount of energy for both motion and maintaining the communication paths between sensors and the base stations; further, base stations may not be allowed to move in some scenarios (e.g., they are connected via wires to the Internet). In the mobile data collector-based approaches [29], [30], [31], [32], [33], a set of mobile data collectors traverse the network periodically to collect the data generated and buffered at sensors, while in the rendezvous-based approaches [34], sensors send their data to designated rendezvous nodes, which are visited by mobile data collectors periodically for data collection.

The Traveling Salesman Problem [35], [36] and the Vehicle Routing Problem with Time Windows [9], [10], which are closely related to our global-tier scheduling, have been extensively studied in the operational research.

9 Conclusion

In this paper, we proposed an NRR strategy and an adaptive rendezvous-based two-tier scheduling scheme to meet the challenges of designing an efficient WSN for long-term tasks. To the best of our knowledge, we are the first to propose the NRR strategy to combat the challenging problem.

In the future, we plan to study the impact of various realistic issues on our design and explore other design choices for the NRR System. Especially, we aim to improve our schemes in the following aspects: first, the current model assumes all sensors to consume a fixed amount of energy δ in each phase. In reality, based on sensor's workload and other factors, different sensors may consume energy at different rates, and a sensor may consume energy at different rates at different phases. The dynamics will be studied in our future work. Second, in our schemes, the MR always reclaims the same number of sensors as it redeploys when it visits a group. We plan to study how our schemes are affected if the number of reclaimed sensors is different from the number of redeployed ones.

Acknowledgments

A preliminary version of this work appears on the proceedings of the Sixth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2009) [12]. The work was partially supported by US National Science Foundation (NSF) grants CNS-0831874 and CNS-0831906.

References

[1] K. Zeng, K. Ren, W. Lou, and P.J. Moran, "Energy-Aware Geographic Routing in Lossy Wireless Sensor Networks with Environmental Energy Supply," *Proc. Int'l Conf. Quality of Service in Heterogeneous Wired/Wireless Networks (QShine '06),* pp. 8-17, 2006.

[2] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design Considerations for Solar Energy Harvesting Wireless Embedded Systems," *Proc. Int'l Symp. Information Processing in Sensor Networks (IPSN '05),* pp. 457-462, 2005.

[3] A. Kansal, J. Hsu, M.B. Srivastava, and V. Raghunathan, "Harvesting Aware Power Management for Sensor Networks," *Proc. Ann. Design Automation Conf. (DAC '06),* pp. 651-656, 2006.

[4] A. Kansal and M.B. Srivastava, "An Environmental Energy Harvesting Framework for Sensor Networks," *Proc. Int'l Symp. Low Power Electronics and Design (ISLPED '03),* pp. 481-486, 2003.

[5] A. Kansal, D. Potter, and M.B. Srivastava, "Performance Aware Tasking for Environmentally Powered Sensor Networks," *Proc. ACM SIGMETRICS,* pp. 223-234, 2004.

[6] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J.A. Stankovic, T.F. Abdelzaher, J. Hui, and B. Krogh, "Vigilnet: An Integrated Sensor Network System for Energy-Efficient Surveillance," *ACM Trans. Sensor Networks,* vol. 2, no. 1, pp. 1-38, 2006.

[7] S. Yoon, C. Veerarittiphan, and M.L. Sichitiu, "Tiny-Sync: Tight Time Synchronization for Wireless Sensor Networks," *ACM Trans. Sensor Networks,* vol. 3, no. 2, pp. 8-40, 2007.

[8] Q. Li and D. Rus, "Global Clock Synchronization in Sensor Networks," *IEEE Trans. Computers,* vol. 55, no. 2, pp. 214-226, Feb. 2006.

[9] M. Desrochers, J. Lenstra, M. Savelsbergh, and F. Soumis, *Vehicle Routing with Time Windows: Optimization and Approximation,* pp. 65-84. Dept. of Operations Research and System Theory, Centrum voor Wiskunde en Informatica, 1987.

[10] "The vrp Web," http://neo.lcc.uma.es/radi-aeb/WebVRP/, 2007.

[11] J.K. Lenstra and A.H.G.R. Kan, "Complexity of Vehicle Routing and Scheduling Problems," *Networks,* vol. 11, no. 2, pp. 221-227, 1981.

[12] B. Tong, G. Wang, W. Zhang, and C. Wang, "Node Reclamation and Replacement for Long-Lived Sensor Networks," *Proc. Ann. IEEE Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON '09),* pp. 592-600, 2009.

[13] T. Liebling, D. Naddef, and L.A. Wolsey, "On the Capacitated Vehicle Routing Problem," *Math. Programming,* vol. 94, nos. 2/3, pp. 343-359, 2003.
[14] M.I. Brownfield and N.J. Davis, "Symbiotic Highway Sensor Network," *Proc. Vehicular Technology Conf.,* pp. 2701-2705, 2005.
[15] B. Liu, O. Dousse, J. Wang, and A. Saipulla, "Strong Barrier Coverage of Wireless Sensor Networks," *Proc. Int'l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc '08),* pp. 411-420, 2008.
[16] L. Zhang, X. Zhou, and Q. Cheng, "Landscape-3D; A Robust Localization Scheme for Sensor Networks over Complex 3D Terrains," *Proc. IEEE Conf. Local Computer Networks (LCN '06),* pp. 239-246, 2006.
[17] S. Du, A.K. Saha, and D.B. Johnson, "RMAC: A Routing-Enhanced Duty-Cycle Mac Protocol for Wireless Sensor Networks," *Proc. IEEE INFOCOM,* pp. 1478-1486, 2007.
[18] F. Wang and J. Liu, "Duty-Cycle-Aware Broadcast in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* pp. 468-476, 2009.
[19] W. Wang, V. Srinivasan, and K. Chua, "Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks," *Proc. ACM MobiCom,* pp. 270-283, 2005.
[20] J. Luo and J.-P. Hubaux, "Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks," *Proc. IEEE INFOCOM,* pp. 1735-1746, 2005.
[21] M. Rahimi, H. Shah, G.S. Sukhatme, J. Heidemann, and D. Estrin, "Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA '03),* pp. 19-24, 2003.
[22] C.M. Vigorito, D. Ganesan, and A.G. Barto, "Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks," *Proc. Ann. IEEE Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON '07),* pp. 21-30, 2007.
[23] C. Alippi and C. Galperti, "An Adaptive System for Optimal Solar Energy Harvesting in Wireless Sensor Network Nodes," *IEEE Trans. Circuits and Systems I,* vol. 55, no. 6, pp. 1742-1750, July 2008.
[24] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Autonomous Deployment and Repair of a Sensor Network Using an Unmanned Aerial Vehicle," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA '04),* pp. 3602-3608, 2004.
[25] L. Filipe, M. Augusto, L. Ruiz, A. Alfredo, D. Ceclio, and A. Fernandes, "Efficient Incremental Sensor Network Deployment Algorithm," *Proc. Brazilian Symp. Computer Networks,* pp. 3-14, 2004.
[26] Y. Mei, C. Xian, S. Das, Y.C. Hu, and Y.-H. Lu, "Sensor Replacement Using Mobile Robots," *Computer Comm.,* vol. 30, no. 13, pp. 2615-2626, 2007.
[27] W. Hu, V. Tran, N. Bulusu, C. Chou, S. Jha, and A. Taylor, "The Design and Evaluation of a Hybrid Sensor Network for Cane-Toad Monitoring," *Proc. Int'l Symp. Information Processing in Sensor Networks (IPSN '05),* pp. 503-508, 2005.
[28] E. Ekici, Y. Gu, and D. Bozdag, "Mobility-Based Communication in Wireless Sensor Networks," *IEEE Comm. Magazine,* vol. 44, no. 7, pp. 56-62, July 2006.
[29] 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,* vol. 1, nos. 2/3, pp. 215-233, 2003.
[30] A.A. Somasundara, A. Ramamoorthy, and M.B. Srivastava, "Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines," *Proc. IEEE Int'l Real-Time Systems Symp. (RTSS '04),* pp. 296-305, 2004.
[31] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C. Lee, "Partitioning Based Mobile Element Scheduling in Wireless Sensor Networks," *Proc. Ann. IEEE Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON '05),* pp. 386-395, 2005.
[32] O. Tekdas, J. Lim, A. Terzis, and V. Isler, "Using Mobile Robots to Harvest Data from Sensor Fields," *IEEE Wireless Comm.,* vol. 16, no. 1, pp. 22-28, Feb. 2009.
[33] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, "Data Muling over Underwater Wireless Sensor Networks Using an Autonomous Underwater Vehicle," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA '06),* pp. 2091-2098, 2006.
[34] D. Jea, A.A. Somasundara, and M.B. Srivastava, "Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks," *Proc. Int'l Conf. Distributed Computing in Sensor Systems (DCOSS '05),* pp. 244-257, 2005.
[35] D.L. Applegate, R.E. Bixby, V. Chvatal, and W.J. Cook, *The Traveling Salesman Problem: A Computational Study.* Princeton Univ. Press, 2007.
[36] S. Arora, "Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and Other Geometric Problems," *J. ACM,* vol. 45, no. 5, pp. 753-782, 1998.

Bin Tong received the BE degree from Xi'an Jiaotong University, China, and the MS degree from East-China Institute of Computing Technology, China, both in computer science. He received the PhD degree in computer science from Iowa State University in 2009. He is currently with Microsoft. His research areas lie in wireless sensor networks and distributed systems. He is a member of the IEEE.

Guiling (Grace) Wang received the BS degree from Nankai University, China. She received the PhD degree in computer science and engineering and a minor in statistics from Pennsylvania State University in May 2006. She joined the New Jersey Institute of Technology as an assistant professor after that. Her research interests include distributed systems, wireless networks, and mobile computing with a focus on wireless sensor networks. She is a member of the IEEE.

Wensheng Zhang received the BS degree from Tongji University, Shanghai, China, and the MS degree from Chinese Academy of Sciences. He received the PhD degree in computer science from the Pennsylvania State University in 2005. Since then, he has been with the Department of Computer Science at Iowa State University as an assistant professor. His research interests are wireless networks and network security. His research is sponsored by US National Science Foundation (NSF). He is a member of the IEEE and the ACM.

Chuang Wang received the BE and ME degrees in computer science from Xi'an Jiaotong University, in 2003 and 2006, respectively. He is working toward the PhD degree at Iowa State University. His research interests include privacy and security issues in wireless networks.