A Mobile Data Gathering Framework for Wireless Rechargeable Sensor Networks with Vehicle Movement Costs and Capacity Constraints

Cong Wang, Ji Li, Fan Ye, Yuanyuan Yang

Dept. of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA

***Abstract*—Several recent works have studied mobile vehicle scheduling to recharge sensor nodes via wireless energy transfer technologies. Unfortunately, most of them overlooked important factors of the vehicles' moving energy consumption and limited recharging capacity, which may lead to problematic schedules or even stranded vehicles. In this paper, we consider the recharge scheduling problem under such important constraints. To balance energy consumption and latency, we employ one dedicated data gathering vehicle and multiple charging vehicles. We first organize sensors into clusters for easy data collection, and obtain theoretical bounds on latency. Then we establish a mathematical model for the relationship between energy consumption and replenishment, and obtain the minimum number of charging vehicles needed. We formulate the scheduling into a Profitable Traveling Salesmen Problem that maximizes profit - the amount of replenished energy less the cost of vehicle movements, and prove it is NP-hard. We devise and compare two algorithms: a greedy one that maximizes the profit at each step; an adaptive one that partitions the network and forms Capacitated Minimum Spanning Trees per partition. Through extensive evaluations, we find that the adaptive algorithm can keep the number of nonfunctional nodes at zero. It also reduces transient energy depletion by 30-50% and saves 10-20% energy. Comparisons with other common data gathering methods show that we can save 30% energy and reduce latency by two orders of magnitude.**

***Index Terms*—Wireless rechargeable sensor networks, perpetual operations, mobile data collection, recharge scheduling, adaptive network partitioning.**

I. Introduction

Wireless charging has opened up a new dimension to power Wireless Sensor Networks (WSNs) and these networks are referred as *Wireless Rechargeable Sensor Networks* (WRSNs) [4]–[17]. Compared to environmental energy harvesting techniques, where sensors scavenge energy from ambient sources such as solar, wind and thermal, which may not always be available, wireless charging provides a reliable energy source without wires or plugs. For high charging efficiency, *charging vehicles* equipped with resonant coils that can move close to nodes are usually adopted [11]–[17]. Recharge sequences are calculated such that nodes are recharged before energy depletion. Ideally, the lifetime of a WRSN can be extended to infinitely long for *perpetual operations*.

However, most of the previous works have ignored the moving energy consumption of the charging vehicle and its limited charging capacity. These simplifications may lead to serious problems in reality. First, they may cause impractical schedules where charging vehicles deplete their energy, become stranded and unable to return to the base station. The network would eventually use up energy and stop operation completely. Second, they tend to overestimate the vehicle's recharge capability and nodes' lifetimes. Real vehicles have limited battery capacity. They have to spend time returning to the base station for battery replacement and cannot keep recharging nodes continuously. Third, they may result in inefficient recharge scheduling and node selection. They may choose nodes faraway simply because they have lower energy levels, and subsequently vehicles travel back-and-forth over long distances, wasting significant amounts of energy.

For WRSNs, energy replenishment cannot be considered separately from energy consumption patterns, which rely on how data is gathered in the network. Previous works in [14], [17] simply utilize a static data sink to gather packets over multi-hops. It is subject to the infamous energy hole problem [3] where nodes near the base station consume energy and deplete batteries much faster, causing service interruptions. A single vehicle that gathers data and charges nodes simultaneously [15] can mitigate the problem. However, it causes high data collection latency due to the non-negligible battery recharge time. A battery requires nontrivial recharge time (e.g., 30 to 90 min) whereas gathering data takes only a few minutes (e.g., 1.6 min for transmitting 3 MBytes at 250 kbps). Thus the waiting time for completing recharge increases dramatically when more nodes need recharge. The gathered data would inevitably experience long latency and may be of little value when delivered to the base station. We propose a comprehensive framework that solves both data collection and recharge scheduling problems. The framework can be applied to many application scenarios such as environmental monitoring, target surveillance and disaster relief. A mobile vehicle can collect and deliver data to the base station in such infrastructureless ad hoc networks. At the same time, mobility enables charging vehicles to move around to replenish sensor nodes' energy around the network.

To eliminate the entanglement between recharging and latency, we employ a separate, dedicated *data gathering vehicle*. Thus the data latency only depends on the mobility pattern (e.g., dispatching frequency, number of stops, speed) of this vehicle. This avoids long latency caused by slow recharging processes [15]. To prevent stopping at every node thus prolonging the tour length and latency, we let nodes form clusters and forward data to cluster heads. Thus only stops at these cluster heads are needed. A series of interesting questions arise in this new scheme. First, what should be the appropriate cluster size such that all nodes are covered while there are not too many clusters causing long latency? Second, what is the minimum number of charging vehicles to cover all the nodes given a bounded cluster size? To answer these questions, we establish a mathematical model for the energy neutral condition to characterize the tradeoff between data collection latency and the number of charging vehicles, both related to the cluster size. A small cluster size leads to more stops, thus higher latency. In the extreme case of single-hop clusters, the vehicle has to traverse through every other node to obtain all the data. A large cluster size reduces latency, but incurs more relaying traffic and more energy consumption. Our model successfully quantifies such trade-offs.

Next, we consider charging vehicles' limited battery capacity and their moving energy consumptions in recharge scheduling. We maximize *recharge profit* (i.e., the recharged energy less the traveling cost), while meeting nodes' battery deadlines and vehicles' capacity constraints. These constraints bring us new challenges. On one hand, recharging nearby nodes reduces a vehicle's moving cost. On the other hand, faraway nodes, not just nearby ones, need recharge once in a while. We have to balance between the need to recharge the whole network and the desire to minimize the traveling cost. In particular, we need to answer the following questions: How to schedule charging vehicles so they will not waste energy traveling back and forth over long distances? Which nodes a charging vehicle should select to ensure it has enough energy to return, and in what orders so as to meet nodes' battery deadlines? We formulate the recharge scheduling problem into an optimization of *Profitable Traveling Salesmen Problem with Capacity and Battery Deadline Constraints*, which was studied before but has only computationally

intensive solutions.

We propose two efficient algorithms. The first is a simple *Greedy Algorithm* that maximizes a charging vehicle's profit at each step. However, it may lead to long traveling distances. We further propose a three-step *Adaptive Algorithm*. After collecting recharge requests, it partitions the network into several regions using the K-means algorithm [35]. Each charging vehicle is assigned a region and its movements are confined within the region, so long-distance travels are avoided. Then each charging vehicle works independently to construct *Capacitated Minimum Spanning Trees* in its designated region where edges in the tree have the minimum traveling cost. This ensures that the charging vehicle's capacity is not exceeded so it can return to its starting position. Finally, the algorithm performs route improvements to meet nodes' battery deadlines. It categorizes nodes according to their lifetimes. An initial route containing nodes that do not need prioritized recharge is first constructed using Traveling Salesmen Problem algorithms. Then it inserts nodes that need prioritized recharge into the route while ensuring each insertion retains time feasibility of the whole recharge sequence.

The contributions are summarized as follows. First, we point out limitations in the existing works on important issues of data latency, vehicle's moving cost, recharge capacity, and their impact on existing recharge scheduling algorithms. We establish a mathematical model to quantify the relationship between data latency and the number of charging vehicles needed. We also present several theoretical results such as node lifetime and adaptive recharge thresholds. Second, we formulate recharge optimization into a Profitable Traveling Salesmen Problem with Capacity and Battery Deadline constraints, and propose two algorithms. The Adaptive Algorithm takes a systematic approach to capture all constraints in the problem. Finally, we conduct extensive simulations comparing the two proposed algorithms. Although we are not able to prove approximation bounds for the Adaptive Algorithm theoretically, simulations show that it is only 1.06 to the optimal solutions and saves an additional 8% on vehicle's moving energy compared to the weighted-sum algorithm in [12]. Moreover, when the number of charging vehicles is sufficient, the Adaptive Algorithm can keep all the nodes alive at all times. Compared to the Greedy Algorithm, the Adaptive Algorithm can reduce nonfunctional nodes by 30-50% while saving 10-20% energy on charging vehicles. We validate our theoretical results and justify the system cost, data latency of our framework compared to other schemes. To the best of our knowledge, this is the first work to explore recharge schedules when both charging vehicles' energy and dynamic sensor battery deadlines are considered. This is also the first work that provides a mathematical model to calculate the minimum number of charging vehicles where detailed communication energy consumption is considered.

The rest of the paper is organized as follows. Section II presents literature reviews of the previous works. Section III outlines the framework, network components and assumptions. Section IV describes the main design of low latency mobile data collection. A mathematical model with a set of theoretical results are derived in Section V. Section VI formalizes the recharge optimization problem and proposes two algorithms. Finally, Section VII provides the evaluation results, Section VIII discusses possible improvements and Section IX concludes the paper.

II. Related Works

A. Radiation-based Wireless Charging

Applications of radiation-based wireless charging have grown rapidly from infancy to maturity recently. Popular commercial products from Powercast [2] can provide energy to nodes in a few meters. Extensive efforts applying the technology to renovate traditional battery-powered WSNs are sought in [4]–[9]. In [4], impact from wireless charging technology on WSNs is studied based on Powercast device models; the sensor deployment and routing problems are solved by new heuristic algorithms. In [5], a greedy algorithm with complexity $\mathcal{O}(k^2k!)$ (k is the number of nodes) was designed to find a recharge sequence to maximize the lifetimes of sensor nodes using Powercast chargers [2]. Although energy of mobile chargers is considered in [5], no step was taken to minimize the traveling energy of the chargers. In [6], a joint routing and wireless charging scheme is proposed to improve network utilization and prolong network lifetime. Similarly, in [7], deployment problems of wireless chargers are studied to extend network lifetime. Another problem of using sensors' battery recharge times for localization is studied in [8]. In [9], safety issues using radiation-based wireless charging are studied. The problem is formulated into a placement problem to guarantee no location is exposed to electromagnetic radiation above a threshold. In [10], a similar problem to optimize the amount of "useful" energy under safety concerns is formulated. In general, these works utilize commercial radiation-based wireless charging products to power sensor nodes.

However, a limitation of this technique is imposed by Federal Communication Commission's (FCC) regulatory maximum effective isotropic radiated power (EIRP) of 4W [18]. Omnidirectional emitting patterns may further exacerbate charging efficiencies as the electromagnetic energy attenuates rapidly over distances. As a result, it can only support low-power, infrequent sensing applications such as temperature reading and is unable to power nodes with more complicated sensing missions, e.g. imaging, video surveillance, tracking, etc. For this reason, in the rest of this paper, we mainly focus on resonant-based wireless charging.

B. Resonant-based Wireless Charging

In contrast to radiation-based technique, resonant-based wireless charging can deliver high amounts of energy at high efficiency [11]–[16]. In [11], batteries can be partially charged and various recharging schemes to traverse the sensing field are explored. In [12], [13], a real-time energy information gathering protocol is proposed to obtain accurate energy status of the network. An on-line algorithm is devised to schedule multiple vehicles to recharge sensor nodes. In [14], a near-optimal solution that dispatches one vehicle to recharge all sensor nodes is provided. However, data is collected by a static data sink, which is less energy efficient. Upon realizing this problem, Zhao et. al [15] use a single vehicle for both wireless charging and data collection to achieve higher efficiency. An algorithm that selects recharging nodes is first proposed followed by a system-wide optimization to maximize the network utility. In [16], a similar approach uses a mobile base station to process data immediately without latency. It requires mobile base station to possess intensive computational capabilities for processing and dissemination of gathered data. Designing such mobile entities would incur much higher manufacturing cost. Although some previous works accounted for charging vehicle's battery energy [5], their strategy is to simply direct the vehicle back to the base station when it depletes energy. In other words, they just passively react upon energy depletion; they do not proactively optimize the recharge schedule under limited energy resources. In contrast, we take a vehicle's recharge capacity and moving cost into problem formulations, and consciously optimize the recharge schedule such that the limited resources are best utilized.

C. Mobile Data Gathering

How data is gathered determines energy efficiency and data latency in the network. Mobile data gathering has been studied extensively [19], [20]. In [19], path-planning algorithms are proposed

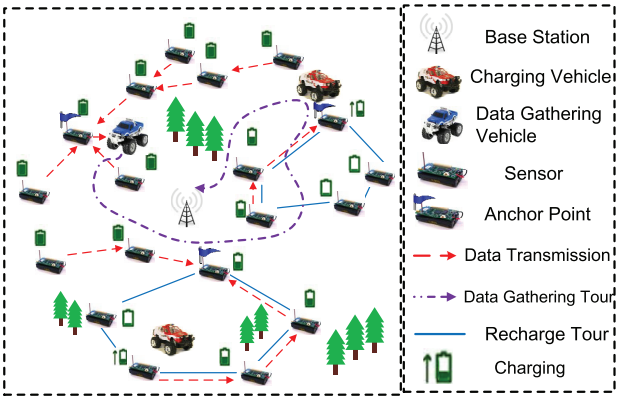


Fig. 1. Illustration of the network architecture and components.

for mobile collectors to collect data from sensors through single or multi-hop relays within a time constraint. In [20], mobile relays are used for relaying packets from energy-rich nodes to normal nodes, and a joint mobility and routing algorithm is proposed to extend network lifetime. For WRSN, previous works either uses static data sink [14], [17], which is less energy efficient, or combine data gathering and wireless charging on a single vehicle [15], which incurs high latency. To achieve a balance, we employ a dedicated data gathering vehicle to overcome these drawbacks.

Scheduling mobile data gathering vehicles is studied in [21], [22]. In [21], several scheduling methods are proposed to dispatch vehicles so that no buffer overflow could occur on sensor nodes. In [22], the network is partitioned into different sectors based on nodes' buffer overflow times. A 2D-tree method further partitions using location information within sectors. To guarantee no buffer overflow, the minimum traveling speed of the vehicle is found. However, such algorithms may not be applied to WRSN directly. First, vehicles need to stop and recharge nodes, which takes significantly longer time than data transmission. Thus vehicles cannot perform data collection continuously as assumed in these algorithms. Second, they do not consider vehicles' traveling costs. Thus a node can be visited repeatedly in short intervals, incurring extra energy consumption that should be avoided.

III. Preliminaries

In this section, we present an overview of the components, network model and assumptions.

A. Network Components

Fig. 1 gives a pictorial illustration of the network. Sensory data is generated at *normal nodes* and aggregated at *anchor points* (i.e., cluster heads) in a multi-hop fashion. A *data gathering vehicle* traverses the sensing field periodically and stops at *anchor points* to collect data. It uploads the collected data to the *base station* at the end of each data collection tour. The base station also provides basic maintenance of the network by offering battery replacement. It can be commanded by network administrators remotely to perform computations such as network partitioning in the Adaptive Algorithm proposed later.

Meanwhile, a fleet of *charging vehicles* query the network for energy information using the mechanism introduced in [12]. The charging vehicles send those queries periodically, make recharge decisions (i.e., which nodes to recharge, in which order) and recharge nodes accordingly. Once a charging vehicle fulfills all requests, it sends out a query to see whether there is new energy request. Both types of vehicles return to the base station and have their own batteries replaced when their energy is low.

TABLE I
List of Notations

Notation	Definition
N_s	Number of sensor nodes
L	Side length of squared sensing field
c	Number of clusters in the network
k	Cluster size in terms of communication hop count
m	Number of charging vehicles
d_r	Transmission range of sensor nodes
e_t, e_r	Energy consumptions for transmitting and receiving a packet
e_c	Energy consumption of charging vehicle while moving
λ	Average packet rate of Poisson distributed traffic
T_c	Data collection period
B	Data uploading bit rate
C_s	Battery capacity of sensor nodes
C_h	Battery capacity of charging vehicles
T_r	Recharge time of sensor's battery
v	Moving speed of vehicles

B. Network Model and Assumptions

We assume a number of N_s sensor nodes are uniformly randomly scattered in a square sensing field with side length L. Node density of the network is $\rho = \frac{N_s}{L^2}$. In this paper, we focus on event-driven sensing applications and assume events occur at every location with equal probability, spatially and temporally independent of each other. Thus, the data generation process can be modeled as a Poisson process with average rate λ [25]. All sensors transmit at the same power level with fixed transmission range d_r. The energy consumed for transmitting/receiving a packet of length l, denoted by e_t, e_r, is modeled as in [26], i.e., $e_t = (e_1 d_r^\alpha + e_0)l$, where e_1 is the loss coefficient per bit, α is the path loss exponent (usually from 2 to 4) and e_0 is energy consumed on sensing, coding, modulations. In this paper, we use $e_0 = 50 \times 10^{-8}$ J/bit, $e_1 = 10 \times 10^{-8}$ J/bit, $\alpha = 4$.

The network is split into a number c clusters. A cluster is formed in a way such that the maximum hop count from a node to the *anchor point* (cluster head) is k. When a node falls within k-hops of multiple anchor points, it will join the cluster with the least number of hops. A *data gathering vehicle* starts from the base station every T_c time period, stops at anchor point location i for time t_i to gather all sensed data and returns to the base station after all anchor points have been visited. The dispatch interval T_c is greater than the duration of the data gathering tour. The data gathering vehicle visits anchor point locations directly to minimize transmission energy consumption on these nodes. The transmission bit rate is B.

There are also m *charging vehicles* working together to replenish sensor batteries. A number of nodes are selected for a vehicle to form its recharge set. If a node cannot survive the time needed to recharge all the other nodes in the set, it needs *prioritized recharge* (i.e., it should be charged earlier in the recharge sequence). Charging vehicles bring sensor batteries from zero to full capacity C_s in T_r time which is governed by battery characteristics (e.g., for a Panasonic Ni-MH AAA battery [23] of battery capacity C_s = 780 mAh and T_r = 78 min.). All the vehicles are equipped with high-capacity batteries of C_h capacity and consume at e_c J/m while moving at speed v m/s. In this paper, we have made the following assumptions: 1) we assume the energy consumption during transmission and reception of a packet is equivalent ($e_r \approx e_t$); 2) the vehicles have positioning systems and know their locations; 3) the locations of all the sensor nodes are known to the vehicles (e.g., through a one-time effort during network initialization). Finally, important notations used in this paper are summarized in Table I.

IV. Low Latency Mobile Data Collection in WRSNs

We now study how to minimize data collection latency given k-hop clusters. To minimize delay, it is desirable to have the data gathering vehicle stop at fewer anchor points. To ensure all data can be collected, the k-hop coverage areas of these anchor points should

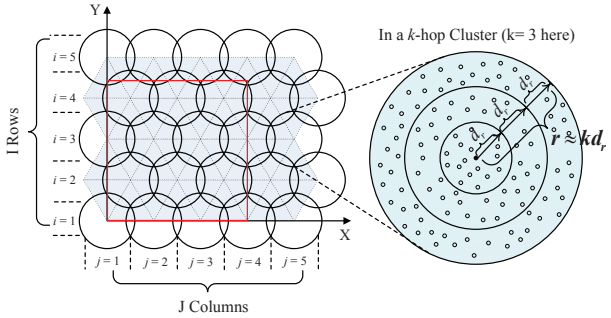


Fig. 2. Example of equilateral triangular tessellation of clusters covering a sensing field with hop count $k = 3$.

collectively cover the entire network. The delay mainly depends on three variables: sum of stopping time at anchor points, traveling time through all anchor points and data uploading time to the base station.

The stopping time at each anchor point depends on the amount of data generated during two consecutive visits of the data gathering vehicle. The traveling time depends on the number of anchor points and vehicle's speed. Hence, let us first determine the number of anchor points that can cover the entire sensing field in k hops. As studied in [3], a k-hop cluster can be closely approximated by a circle with radius $r = kd_r$ with k coronas as shown in Fig. 2. Then, finding the minimum number of anchor points is equivalent to finding a complete coverage of the sensing field with minimum number of circles of radius r. The problem is closely related to the circle covering problem studied by Kershner [24], which gives the minimum number of circles needed to cover a rectangular region in the following lemma.

Lemma 1: The number of circles c to cover a sensing field with area L^2(L is the side length of the field) has the lower bound of ([24])

$$c > \frac{2\pi\sqrt{3}(L^2 - 2\pi r^2)}{9\pi r^2} \tag{1}$$

Although the exact placement pattern to achieve this lower bound was not given in [24], it has been proved in [19] that the maximum coverage is achieved when we tessellate the sensing area with equilateral triangles of side length $\sqrt{3}kd_r$ and place the centers of circles at the vertices of triangles. However, how to place these clusters in a square sensing field considering the effects of boundaries was not discussed in [19] so we introduce a placement pattern first. For a square sensing field with the origin (0,0) at the left bottom, we place I circles parallel to the x-axis and J circles parallel to the y-axis, then the cartesian coordinates of centers of circles at the i-th row, j-th column are

$$[X_{ij}, Y_{ij}] = \begin{cases} [\sqrt{3}(j-1)r, \frac{3}{2}(i-1)r] \\ i = \{2u+1; \forall u \in \mathbb{Z}\} \\ [\frac{\sqrt{3}}{2}r + \sqrt{3}(j-1)r, \frac{3}{2}(i-1)r] \\ i = \{2u; \forall u \in \mathbb{Z}\} \end{cases} \tag{2}$$

After the deployment pattern has been determined, the number of circles I to cover each row can be calculated as

$$I = \begin{cases} \lfloor \frac{L}{\frac{3}{2}r} \rfloor + 1, \frac{L}{\frac{3}{2}r} - \lfloor \frac{L}{\frac{3}{2}r} \rfloor \leq \frac{1}{2} \\ \lfloor \frac{L}{\frac{3}{2}r} \rfloor + 2, \text{otherwise} \end{cases} \tag{3}$$

The number of circles J to cover each column with an odd index $i = \{2u+1; \forall u \in \mathbb{Z}\}$ is

$$J = \begin{cases} \lfloor \frac{L}{\sqrt{3}r} \rfloor + 1, \frac{L}{\sqrt{3}r} - \lfloor \frac{L}{\sqrt{3}r} \rfloor \leq \frac{1}{2} \\ \lfloor \frac{L}{\sqrt{3}r} \rfloor + 2, \text{otherwise} \end{cases} \tag{4}$$

The number of circles J to cover each column with an even index $i = \{2u; \forall u \in \mathbb{Z}\}$ is

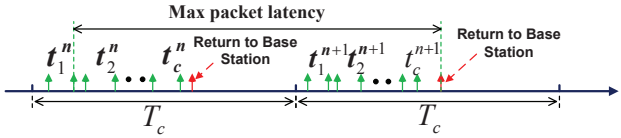


Fig. 3. A timing diagram of two consecutive mobile data gathering tours.

$$J = \begin{cases} \lfloor \frac{L}{\sqrt{3}r} \rfloor, \frac{L}{\sqrt{3}r} - \lfloor \frac{L}{\sqrt{3}r} \rfloor = 0 \\ \lfloor \frac{L}{\sqrt{3}r} \rfloor + 1, \text{otherwise} \end{cases} \tag{5}$$

Fig. 2 shows an example of equilateral triangular tessellation of 14 clusters covering a square sensing field with $k = 3, L = 3\sqrt{3}r$. Compared to the lower bound of $c > 7.97$ obtained from Eq. (1), an additional 6 clusters are needed to cover the boundaries of the field. Given a field length L, the number of clusters c (number of anchor points), coverage of the entire sensing field can be obtained from Eqs. (3), (4) and (5). Then we derive an upper bound of mobile data gathering latency in the following lemma.

Lemma 2: The mobile data gathering latency is bounded by

$$T_d \leq T_c + (c-1)T_s + (\sqrt{2(c-3)}L + 4L)/v \tag{6}$$

where T_d is the data latency, $T_s = F_\lambda^{-1}(\epsilon)\rho r^2 \pi l T_c / B$, $F_\lambda^{-1}(x)$ is the inverse CDF of Poisson distribution with average rate λ, ϵ is a value close to 1 but not equal to 1 (e.g. $\epsilon = 0.99$), v is the vehicle's speed.

Proof: Fig. 3 shows a timing diagram of mobile data gathering. t_i^n is the stopping time at the i-th anchor point during the n-th round of data gathering. We observe that the maximum latency occurs when a packet arrives at the first anchor point in the visiting sequence after the data gathering vehicle has left. Then the packet has to be buffered and wait for another collection period after time T_c, plus sum of stopping time at subsequent anchor points, traveling time to the base station through the rest of anchor points. The maximum stopping time T_s at an anchor point occurs when each node generates at maximum data rate $F_\lambda^{-1}(\epsilon)$. Note that ϵ is a value very close to 1 but not equal to 1 (e.g., $\epsilon = 0.99$) because $F_\lambda^{-1}(1) \to \infty$. For each cluster with a number of $\rho r^2 \pi$ sensors, $T_s = F_\lambda^{-1}(\epsilon)\rho r^2 \pi / B$. Therefore, the sum of stopping time at subsequent anchor points is bounded by $(c-1)T_s$.

To traverse $(c-1)$ nodes, a deterministic upper bound on the shortest tour length was given in [38]. That is, for n points in a rectangle with size $a \times b$, the shortest tour length $s < \sqrt{2(n-2)ab} + 2(a+b)$. Here, $a = b = L$, $n = c-1$, so the upper bound of traveling time is $(\sqrt{2(c-3)}L + 4L)/v$. By summing by this result with maximum stopping time at subsequent anchor points $(c-1)T_s$ and T_c, we have derived an upper bound of mobile data gathering latency. ■

From *Lemma 2*, we can compare the data gathering latency with the combined approach in [15] numerically. For charging vehicles of battery capacity 12Ah of 5V ($C_h = 216KJ$), a recharge tour would take around $\frac{C_h T_r}{C_s} = 32$ hours to finish. This amounts to at least 32 hours waiting time for the data to be delivered to the base station till the vehicle returns to the base station for battery replacement. For our approach, we set $N_s = 500$, $T_c = 60$ mins, $r = 45$ m, $c = 14$, $L = 160$m, $B = 250$ Kbps, $l = 10$ bytes, $\lambda = 3$ and after plug into Eq. (6), we have $T_d \leq 1.65$ hours which is significantly less than the combined approach about an order of magnitude. For further improvement of latency, we can dispatch the data gathering vehicle more frequently by using a small T_c. We will use different T_c to see their average latencies and corresponding upper bounds in the simulations.

V. Number of Charging Vehicles for k-hop WRSN

Having discussed k-hop cluster formation and data latency in our framework, we now analyze the minimum number of charging

vehicles needed to fulfill all energy requests given the number of clusters c obtained from Eqs. (3), (4), (5).

A. Number of Charging Vehicles

In an earlier work [12], we have proposed the energy neutral condition that must hold in a long time period for the perpetual operation of the network,

$$E(T) \leq R(T) + E_0 \tag{7}$$

in which T is a large time, $E(T)$ is the total energy consumption of the network up to T, $R(T)$ is the total energy replenished into the network by the charging vehicles up to T and E_0 is the initial energy of all the sensor nodes. The energy neutral condition states that the energy consumption of all the sensor nodes must be less than or equal to the total energy available in long term. Otherwise, sensor nodes would eventually deplete energy. Note that for the network to function, it is not necessary for the condition to hold at every single moment. In practice, a small fraction of the network may consume more energy in a short time window due to external activities, leading to temporary unbalance between energy consumption and replenishment. As long as there are enough charging vehicles, these nodes will be recharged, and such unbalance is transient, not permanent.

Our objective is to obtain the minimum number of charging vehicles m needed for Eq. (7) to hold. First, we estimate $R(T)$ which is the amount of energy that can be replenished into the network. The maximum recharge capacity of a charging vehicle is achieved when it recharges sensor nodes continuously without any idling time. The longest recharging time for a sensor occurs when a node's energy is brought from zero energy to full capacity which takes T_r time plus the longest moving time between two consecutive sensors in the recharge sequence (moving on the diagonal of the square sensing field). Therefore, in the worst scenario, it takes $\sqrt{2}L/v + T_r$ time to recharge each sensor. Then we can estimate the energy replenished into the network in T time by m charging vehicles,

$$R(T) = \frac{mC_bT}{\sqrt{2}L/v + T_r}. \tag{8}$$

Next, we need to derive $E(T)$ on the left hand side of Eq. (7) which is a random variable. Given the structure of the cluster of radius $r = kd_r$, each corona carries traffic loads from all outer coronas. The number of nodes in the i-th corona, is $N_i = (2i-1)d_r^2\pi\rho$ for $0 < i \leq k$. Since the outmost k-th corona only needs to send out its own data and data is generated independently, the mean of energy consumption at the k-th corona μ_k in time period T is,

$$\mu_k = N_i\lambda T e_t = (2k-1)d_r^2\pi\rho\lambda T e_t \tag{9}$$

For the i-th corona ($0 < i < k$), it carries all the traffic from the outer coronas so the mean energy consumption is,

$$\begin{aligned}\mu_i &= N_i\lambda T e_t + \sum_{j=i+1}^{k} N_j\lambda T(e_t + e_r) \\ &= d_r^2\pi\rho\lambda T\big((k^2-i^2)(e_t+e_r) + (2i-1)e_t\big)\end{aligned} \tag{10}$$

Then we can compute the mean of network energy consumptions $\overline{E(T)}$,

$$\begin{aligned}\overline{E(T)} &= \Big(\sum_{i=1}^{k-1}\big((k^2-i^2)(e_t+e_r) + (2i-1)e_t\big) \\ &+ (2k-1)e_t + k^2(e_t+e_r)\Big)d_r^2\pi\rho\lambda T c \\ &= \left(\big(\frac{2}{3}k^3 - \frac{1}{2}k^2 - \frac{1}{6}k\big)(e_t+e_r) + k^2 e_t\right) d_r^2\pi\rho\lambda T c\end{aligned} \tag{11}$$

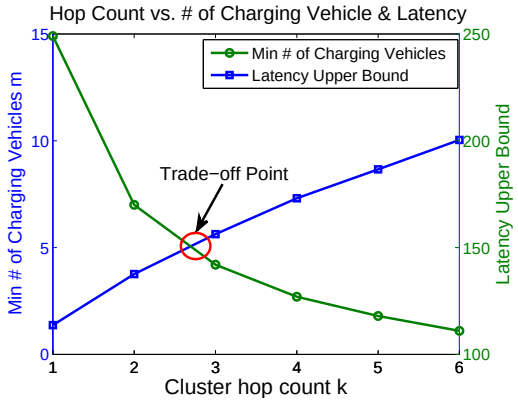


Fig. 4. Trade-off between number of charging vehicles and data collection latency.

Based on the energy neutral condition, by combining $R(T)$ in Eq. (8) and $\overline{E(T)}$ in Eq. (11), we have the following lemma. *Lemma 3:* The probability for the energy neutral condition to hold is

$$P_{op} = \Phi\left(\frac{R(T) + E_0 - \overline{E(T)}}{\sqrt{\overline{E(T)}}}\right) \tag{12}$$

where $R(T)$ and $\overline{E(T)}$ are obtained in Eq. (8) and Eq. (11), respectively. $\Phi(\cdot)$ denotes the Cumulative Distribution Function of the Normal distribution.

Proof: Energy consumption of a cluster can be described by the sum of independent Poisson variables over T. When T is observed over a long time period, we can use the *Central Limit Theorem* to approximate Poisson distribution by a Normal distribution $\mathcal{N}(\overline{E(T)}, \overline{E(T)})$(the mean and variance of a Poisson distribution is the same) [28]. ■

From Lemma 3, we immediately get the following Proposition.

Proposition 1: The minimum number of charging vehicles required to achieve perpetual operation is

$$m = \left\lceil \frac{(\Phi^{-1}(\epsilon)\sqrt{\overline{E(T)}} + \overline{E(T)} - E_0)(\sqrt{2}L/v + T_r)}{C_s T} \right\rceil \tag{13}$$

where $\Phi^{-1}(\epsilon)$ is the inverse cumulative distribution function of Normal distribution, ϵ is a value very close to 1 but not equal to 1.

Proof: Since $\Phi^{-1}(1) \to \infty$, we consider the network achieves perpetual operation with very high probability approaches 1 but not equal to 1, e.g. $\epsilon = 0.99, \Phi^{-1}(0.99) \approx 2.33$. From Eq. (12), we have

$$\frac{\frac{mC_bT}{\sqrt{2}L/v+T_r} + E_0 - \overline{E(T)}}{\sqrt{\overline{E(T)}}} \geq 2.33,$$

after some manipulations we can obtain the minimum number of charging vehicles m needed to satisfy the energy neutral condition. ■

Based on the results from *Proposition 1* and *Lemma 2*, we demonstrate the trade-off between number of charging vehicles and data latency. For $L = 400$ m, we change the number of cluster hop count k and plot the corresponding number of vehicles needed as well as upper bound of data latency in Fig. 4. We can see a trade-off point around $k = 3$. It means when $k = 3$, we can minimize the number of charging vehicles without sacrificing too much from the data collection latency.

B. Estimate Node Lifetime

To devise effective recharging schedules, we need to know how long a sensor node can survive after it has requested for recharge. Such information is vital in making recharge decisions in the next

section. Since a node's energy consumption rate is a random variable and depends on traffic patterns, it is important for each node to know its traffic burden which is determined by the number of hops from base station. This information can be obtained by message propagation from the base station in various routing protocols and adjusted accordingly during operation.

From Eqs. (9) and (10), we know the average traffic rate of a node in the j-th corona ($1 \leq j \leq k$) is, $\lambda_j = \lambda(1 + (k^2 - j^2)/(2j-1))$. Given residual energy E_r, the maximum number of packets the node can transmit is $n = \lfloor \frac{E_r}{(e_t+e_r)} \rfloor$.

Lemma 4: Given a recharge sequence of N nodes in which a node at the j-th corona waiting to be recharged, it will survive time t with probability (lifetime $L_j > t$),

$$P(L_j > t) = 1 - \frac{\gamma(N, \lambda_j t)}{\Gamma(N)}, \tag{14}$$

where $\gamma(\cdot,\cdot)$ and $\Gamma(\cdot)$ are the lower incomplete gamma function and complete gamma function [28], respectively.

Proof: Since sensor nodes are randomly deployed in the field, and the data generation process is independent of each other, the summation of packet interarrival times until the sensor node can no longer transmit packets is the lifetime of the sensor node. Because data generation is Poisson distributed with rate λ_j, the interarrival time of packets is exponentially distributed. It is known that the sum of independently identically distributed exponential variables results a *Gamma distribution* with probability density function

$$f_{L_j}(x) = \lambda_j e^{-\lambda_j x} \frac{(\lambda_j x)^{N-1}}{(N-1)!}, x \geq 0 \tag{15}$$

and the Cumulative Distribution Function of Gamma distribution is

$$P(x < t) = \int_0^t \lambda_j e^{-\lambda_j x} \frac{(\lambda_j x)^{N-1}}{(N-1)!} dx = \frac{\gamma(N, \lambda_j t)}{\Gamma(N)} \tag{16}$$

■

Proposition 2: For the recharge sequence of N nodes, if a node at the j-th corona has probability $\frac{\gamma(N,\lambda_j T_l)}{\Gamma(N)} \approx 0$, $T_l = (N-1)(T_r + \sqrt{2}L/v)$, no matter where the node is placed in the recharge sequence, it will not deplete battery energy before its recharging starts.

Proof: The worst case occurs when the node is placed at the end of the recharge sequence. The longest waiting time to get recharged is $T_l = NT_r + (N-1)\sqrt{2}L/v$ since there are $N-1$ nodes ahead with $\sqrt{2}L/v$ maximum traveling time between two sensor nodes and $\sqrt{2}L$ is the diagonal of the square field. Once $\frac{\gamma(N,\lambda_j T_l)}{\Gamma(N)} \approx 0$, $P(L_j > T_l)$ approaches probability 1 so it is guaranteed to recharge the node before it depletes battery energy. ■

Based on *Proposition 2*, given a recharge sequence, we can calculate the possibility that a node can survive the entire recharging process. This lays the theoretical foundations to solve the recharge scheduling problem in the next section.

C. Adaptive Recharge Threshold

We observe that the difference of energy consumptions between nodes at different locations is mainly caused by data communications. Although the hop count for clusters k should not be too large to avoid the energy hole problem on anchor points, it is inevitable to have higher data traffic in the inner coronas. If all the nodes follows a universally same recharge threshold, it may result some nodes closed to the anchor point nodes to deplete energy very soon and lead to unfair service for nodes with higher consumption rates. To this end, the recharge thresholds should be made adaptive and proportional to energy consumption rates at different coronas. In other words, nodes closer to the anchor points should request recharge more frequently than others.

Let $\tau_i (0 < \tau_j < 1)$ denote the recharge thresholds for nodes at the j-th corona. We make the ratio between the recharge thresholds of corona i and j equal to that between their energy consumptions due to data transmission. Assume we have set the recharge threshold of the first corona to be τ_1. The thresholds for other coronas are,

$$\tau_i = \frac{(k^2 - i^2)(e_t + e_r) + e_t(2i-1)}{(k^2-1)(e_t+e_r) + e_t} \tau_1 \approx \frac{2k^2 - (i-1)^2 - i^2}{2k^2 - 1}, \tag{17}$$

where $0 < i < k$. The approximation is taken under the assumption that $e_t \approx e_r$. To illustrate Eq. (17), e.g. $k = 5$, after τ_1 is set, we obtain $\tau_2 = \frac{45}{49}\tau_1$, $\tau_3 = \frac{37}{49}\tau_1$, $\tau_4 = \frac{25}{49}\tau_1$ and $\tau_5 = \frac{9}{49}\tau_1$.

VI. Capacitated Multi-Vehicle Recharge Problem with Battery Deadlines

During operation, the charging vehicles query sensors for recharge and they usually engage in multiple recharge tasks at different locations. In this section, we study a Capacitated Multi-Vehicle Recharge Problem with Battery Deadlines (CaMP-BaD) and consider practical constraints from real sensing applications. The first challenge is the constant changes (i.e., decrease) of charging vehicles' energy due to moving and recharging sensor nodes. The recharge route should be planned carefully to reflect charging vehicles's current energy status and traveling costs to nodes' locations. The second challenge is the nonuniform energy consumption due to data transmissions. Some nodes consume energy at higher rates and should be taken care of more frequently than others to maintain the functionality of the network. The recharge routes should reflect all aforementioned concerns. The difficulty of the problem lies in achieving conflicting goals - the need to keep the whole network running pushes the charging vehicles to recharge as many sensor nodes as possible while the desire to reduce cost means that charging vehicles should minimize traveling distances to save energy cost. Therefore, an ideal solution should achieve a good balance between the two without sacrificing either.

Next we show the recharge problem can be formulated into a *Profitable Traveling Salesmen Problem with Capacity and Battery Deadline constraints* (PTSP). In the Profitable Traveling Salesmen Problem [29], a reward is collected by visiting a city while the objective is to maximize the profit which is defined as the reward minus cost. In our problem, the reward represents the amount of energy that can be replenished into a sensor node and the cost measures the energy cost in traveling to that node's location.

To tackle the problem, we first present a straightforward *Greedy Algorithm* (GA). After realizing that the greedy algorithm might incur extra movements of charging vehicles, we further propose a three-step *Adaptive Algorithm* (AA) through 1) adaptive network partition using K-means algorithm, 2) Capacitated Minimum Spanning Tree (CMST) formation and 3) route improvements using node insertions. By partitioning the network, the charging vehicles are confined in their own regions so traveling back and forth through the entire field is avoided. Then we form CMST for each charging vehicle. The trees indicate which subset of sensor nodes the charging vehicle should select to minimize traveling cost and ensure the total weight of the tree is within the charging vehicle's recharge capacity. After that, we perform route improvements on nodes in CMST to capture sensor nodes' dynamic battery deadlines. Finally, we analyze the complexity of the proposed algorithms.

A. Problem Formulation

The recharge optimization problem can be defined as follows. Given a set of charging vehicles $\mathcal{S} = \{1, 2, \ldots, m\}$ and a set of recharge node list $\mathcal{N} = \{1, 2, \ldots, n\}$, we formulate the CaMP-BaD problem into a PTSP problem. Consider a graph $G = (V, E)$, where

V_i ($i \in \mathcal{N}$) is the location of sensor node i to be visited, and E is the set of edges. We add a vertex V_0^a as the starting position of vehicle a. Each edge E_{ij} is associated with a traveling energy cost c_{ij}, which is proportional to the distance between nodes i and j, c_{0i}^a represents the cost from initial position V_0^a of vehicle a to node i. A charging vehicle a has recharge capacity C_a ($\leq C_h$) that determines the maximum number of nodes it can recharge before it goes back to the base station for its own battery replacement. Different charging vehicles might have different recharge capacities during the run. Each sensor node i has lifetime L_i and demand (reward) for energy recharge r_i (demand equals the total battery capacity of a sensor node minus its residual energy). A_i specifies the arrival time for a vehicle at sensor node i.

We introduce two decision variables x_{ij}^a for edge E_{ij} and y_{ia} for vertex V_i. The decision variable x_{ij}^a is 1 if an edge is visited by vehicle a, otherwise it is 0. The decision variable y_{ia} is 1 if and only if node i is served by vehicle a, otherwise it is 0. u_i is the position of vertex i in the path. Our objective is to maximize the total amount of energy recharged minus total traveling energy cost of the charging vehicles while ensuring the recharge capacities of charging vehicles are not exceeded and no sensor node depletes battery energy.

$$\mathbf{P1}: \quad \max \left\{ \sum_{a=1}^{m} \sum_{i=1}^{n} r_i y_{ia} - \sum_{a=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}^a - \sum_{a=1}^{m} \sum_{i=1}^{n} c_{0i}^a x_{0i}^a \right\} \tag{18}$$

Subject to

$$\sum_{j=1}^{n} x_{0j}^a = 1; a \in \mathcal{S} \tag{19}$$

$$\sum_{i=1}^{n} x_{ik} = \sum_{j=1}^{n} x_{kj} = 1; k \in \mathcal{N} \tag{20}$$

$$\sum_{i=1}^{n} r_i y_{ia} + \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}^a + \sum_{i=1}^{n} c_{0i}^a x_{0i}^a \leq C_a; a \in \mathcal{S} \tag{21}$$

$$\sum_{a=1}^{m} y_{ia} = 1; i \in \mathcal{N} \tag{22}$$

$$A_i \leq L_i; i \in \mathcal{N} \tag{23}$$

$$x_{ij}^a \in \{0,1\}; i,j \in \mathcal{N}, a \in \mathcal{S} \tag{24}$$

$$y_{ia} \in \{0,1\}; i \in \mathcal{N}, a \in \mathcal{S} \tag{25}$$

$$1 \leq u_i \leq n_r; i \in \mathcal{N} \tag{26}$$

$$u_i - u_j + (n_r - m) x_{ij} \leq n_r - m - 1; i,j \in \mathcal{N}, i \neq j \tag{27}$$

In the above formulation, constraint (19) states that the recharge tour for each charging vehicle starts at initial position 0. Constraint (20) ensures the connectivity of the path and every vertex is visited at most once. Constraints (21) and (22) guarantee the vehicle's capacity is not violated and each vertex is visited by only one charging vehicle. Constraint (23) guarantees the arrival time of the charging vehicle is within each sensor's residual lifetime. Constraints (24) and (25) impose x_{ij} and y_{ia} to be 0-1 valued. Constraints (26) and (27) eliminate the subtour in the planned route, which is formulated according to [30]. The classic TSP with Profits can be considered as a special case of CaMP-BaD with unlimited capacity and unspecified deadlines. Since TSP with Profits is well known to be NP-hard [29], CaMP-BaD is also NP-hard.

A direct solution to the CaMP-BaD is difficult to obtain and rare in existing literature. Hence, we review some literature that has partially solved the problem. In [31], a survey of different approaches to TSP with profits is presented. Lagrangian decomposition method and approximated algorithms developed based on existing solutions can provide solutions very closed to optimality. However, adding capacity (Eq. (21)) and time (Eq. (23)) constraints makes the problem more complicated. A great deal of research efforts on these two constraints are devoted in the context of Vehicle Routing Problem, in which a number of vehicles start from a depot to visit client locations and the objective is to minimize the total traveling cost of the vehicles. In [34], the Capacitated Vehicle Routing Problem where each vehicle has a fixed capacity is considered. Time constraints are studied in Vehicle Routing Problem with Time Windows [32]. A local search algorithm is proposed in [32] based on the k-exchange concept, and reduction of the computation for checking feasibility constraint is also studied. A theoretical approach to obtaining $3 \log n$-approximation algorithm is sought in [33] (n is the number of nodes). However, subroutines from existing solutions visit the node with the smallest deadline last, which contradicts to our problem where such nodes should be serviced earlier.

Due to the nature of our problem, it is not realistic to use standard optimization techniques [31], [34] because these methods deal with datasets of static inputs and the optimization is usually done offline through a one-time effort. In contrast, energy consumption in our framework is probabilistic in nature. A charging vehicle's recharge capacity declines after recharging sensor nodes, so the input to our problem is more dynamic than that most existing solutions have considered. Furthermore, existing algorithms require high computation power that may not be available on charging vehicles. Therefore, we need to design algorithms suitable to our problem context. Next, we present two such algorithms.

B. Greedy Algorithm (GA)

The simplest approach is a greedy algorithm which selects the node with the maximal recharge profit (i.e., recharge reward less traveling cost) for each node selection. After a charging vehicle finishes recharging a node, it picks the next available node with the maximal profit. When the charging vehicle's energy falls below a threshold χ, it returns to the base station for battery replacement and then resumes recharge in the same fashion.

Despite of its simplicity, GA may have some problems in practice. The first problem is that the charging vehicle might move back and forth over long distances, thereby increasing the traveling energy cost. This happens when the node with the maximum profit lies faraway, and the energy efficiency of charging vehicles can deteriorate. Second, because the only consideration is profit, it may not fulfill a recharge request in a fixed time. These observations offer us room for further improvements. To prevent charging vehicles from traveling long distances, we can confine the scope of movements by partitioning the network into several regions adaptively and assigning each charging vehicle to one of the regions. Second, a more sophisticated scheduling method should be developed to capture charging vehicles' capacity as well as sensors' battery deadline constraints. In the next subsection, we will introduce an Adaptive Algorithm (AA) to address the limitations in GA.

C. Recharge by Adaptive Algorithm (AA)

1) Adaptive Network Partitioning: In the first step, the base station requests sensor nodes for energy information periodically using the method in [12]. Then it adaptively partitions the network into m *regions* according to the originating locations of requests. The result of partitions is disseminated to the charging vehicles using long range radio. We utilize the well-known K-means algorithm to perform the partition [35]. Using the K-means algorithm would allow the charging vehicles to adaptively select a subset of nodes with their square sum of distance minimized regarding to the

centroid of each region so the charging vehicle would only move in a confined scope, and most likely with less distances. For each region, our objective is to minimize the intra-region square sum of inter-node distance.

$$S = \sum_{j=1}^{m} \sum_{i=1}^{n_r} \|n_i^{(j)} - \mu^{(j)}\|^2 \tag{28}$$

in which $\|n_i^{(j)} - \mu^{(j)}\|^2$ is the square distance between a recharge node n_i of region j to the region's centroid $\mu^{(j)}$ (computed by taking the mean of x, y coordinates of all the nodes in the region). Now we briefly explain the partitioning process.

Initially, we select a number of m sensor nodes with the minimum lifetime from $\mathcal{N}$ to be the centroid of regions. We assign each node to the closest centroid. After all the nodes have been associated with a centroid, we re-calculate centroid positions taking the average value of x and y coordinates of nodes in the region. This process is repeated until the centroids no longer change. After the partition, the centroid of each region represents a virtual position that has the minimal sum of distances to all the nodes in its region. This position can be used as the starting position for the charging vehicle to recharge nodes in its region.

2) Generating Capacitated Minimum Spanning Tree: In the first step, m regions are generated and each charging vehicle only needs to take care of the nodes in its region. To decide the route to recharge sensor nodes, we need to ensure each charging vehicle's recharge capacity is not exceeded (Eq. (21)). At the same time, we also want to minimize the traveling energy cost for the charging vehicle. These requirements lead to finding Capacitated Minimum Spanning Tree (CMST) [36] where the total sum of demands from nodes does not exceed the charging vehicle's capacity and the minimum traveling energy cost can be found by constructing the minimum spanning tree. In this way, we can ensure sensor nodes closed to each other are placed in the same tree and later covered by the same recharge route.

The exact solution to CMST requires us to go over all possible tree setups and pick the one with the lowest cost, which involves exponential computation. Fortunately, an efficient algorithm by Esau-Williams(EW) can find a suboptimal solution very close to the exact solution in polynomial time [36]. The EW algorithm merges any two subtrees when there is a "saving" in the total cost of two trees.

Nevertheless, there are some limitations of the original EW algorithm when applied for our problem. First, when determining whether two subtrees can be merged, only the demands from sensor nodes are counted whereas the traveling costs on edges are not considered. Second, multiple such trees can be generated. How does the charging vehicle decide which tree to pick? To overcome these limitations, we extend the original EW algorithm. As mentioned earlier, a deterministic upper bound on the shortest tour length is developed as $\sqrt{2(n-2)ab+2(a+b)}$ for a rectangle of side length a, b and n nodes. For the square sensing field with L side length and subtree of n_b nodes, we have a loose upper bound on the traveling cost, $(\sqrt{2(n_b-2)}+2)Le_c$. Second, when multiple trees are generated, we select a tree that maximizes the ratio of total energy demand to traveling cost. In this way, we can exploit limited resources on charging vehicles better and improve energy efficiency of the network.

Next, we explain our extension to the EW algorithm in detail. Each charging vehicle computes CMST independently by iteratively updating a distance matrix. The distance matrix facilitates the computation process by maintaining costs of tree nodes. Let us denote recharge set $\mathcal{N}_a$ with n_a nodes for charging vehicle a ($\bigcup_{a=1}^{m} \mathcal{N}_a = \mathcal{N}_r$). We define trade-off function t_i for each node in its recharge set $\mathcal{N}_a$, $t_i = \min(c_{ij}^{(a)}) - c_{0i}^{(a)}$ and $j \in P_i$, where P_i is the neighboring set of node i, $\min(c_{ij}^{(a)})$ finds the minimum cost from node i to its neighbor j in P_i and $c_{0i}^{(a)}$ is the cost from node i to charging vehicle's starting position (i.e., the root)[1]. The trade-off function evaluates whether it is beneficial to merge subtrees of nodes i and j. A positive t_i indicates that it incurs smaller cost for the charging vehicle to directly travel from the root to node i so merging subtrees of nodes i and j is not preferred. A negative t_i indicates how much it can be saved by connecting subtrees of i and j. Thus the most negative t_i results in the most savings in an iteration.

After t_i has been computed, we search through all trade-offs $t_i(\forall i = 1, \ldots, n_a)$, looking for the minimum trade-off (i.e., the most negative value). Assume t_k is the most negative trade-off and j is k's minimum cost neighbor. To capture charging vehicle's capacity constraint in Eq. (21), if the sum of total demands from the subtrees of k and j plus upper bound of their traveling cost is less than the recharge capacity (which means we can cover the subtrees of k and j under the current recharge capacity), we merge the subtrees of k and j. Since the action of merging k and j has resulted in a lower total traveling cost to k, direct traveling from the root to reach k has higher cost and should be avoid. So we remove the edge from node k to the root by setting $c_{0k}^{(a)}$ in the distance matrix to ∞.

At this point, two subtrees satisfying the recharge capacity with minimum sum of cost have been merged, and we need to update the minimum cost of the newly merged tree to the root. It is done by updating the minimum cost in the distance matrix from the tree to the root by setting the value to $\min(c_{0i}^{(a)})$, where i is the node in the newly merged tree.

On the other hand, if merging subtrees of k and j violates charging vehicle's recharge capacity, we need to restrict any further actions to merge j to k because these two trees cannot be covered by the charging vehicle in a single run. Then we recompute the trade-off function t_k to search for the next neighboring node that results in minimum trade-off until the next valid neighboring node j is found and merged to the existing trees. The iteration continues until all the trade-offs become nonnegative, in other words, no more saving can be made.

After the CMST has been generated, the charging vehicle selects a tree with the maximal ratio of recharge demand to sum of tree's edge cost and utilize the route improvement algorithm to form the final recharge sequence among the tree nodes. After the charging vehicle finishes recharging nodes in a tree, it checks whether its energy falls below a threshold. If so, it returns to the base station for battery replacement. Table II shows the pseudo-code of our extended EW algorithm.

3) Insertion Algorithm for Route Improvement: After the CMST has been obtained, next we want to produce a recharge sequence for nodes such that for each node the charging vehicle arrives before its battery deadline. Let us denote the result from CMST to be a recharge node set $\mathcal{N}_r^{(a)}$ ($\mathcal{N}_r^{(a)} \subseteq \mathcal{N}_a$). Recall that if the condition in Proposition 2 is satisfied, a node can be placed anywhere in the recharge sequence. We call such a set of nodes a *feasible node set* $\mathcal{N}_f^{(a)}$. Otherwise, a node may need prioritized treatment to meet its battery deadline. We denote such a set of nodes as a prioritized set $\mathcal{N}_p^{(a)}$ ($\mathcal{N}_f^{(a)} \cup \mathcal{N}_p^{(a)} = \mathcal{N}_r^{(a)}$).

Intuitively, we first use a Traveling Salesman Problem algorithm (e.g., the $\mathcal{O}(n^2)$ nearest neighbor heuristic algorithm [37], where n is the number of nodes) to find a feasible solution as the initial sequence Ψ for nodes in the feasible set $\mathcal{N}_f^{(a)}$. Then we insert nodes

[1]In order to reduce intra-region traveling cost, we set the centroid output from K-means algorithm to be the root.

TABLE II
EXTENDED ESAU-WILLIAMS ALGORITHM

input: recharging node set $\mathcal{N}_r$, distance matrix $D^{(a)}$, recharge capacity C_a, demand of nodes d_i, $i \in \mathcal{N}_a$.
output: CMST nodes need to recharge.
Initialize $t^{(a)} < 0$, weight of tree, $C^{(a)} = 0$.
while $(t^{(a)} < 0)$
  Find neighbor m_i of i results min cost, $\min_{m_i} D^{(a)}(i, m_i)$.
  Compute trade-off value list $t_i^{(a)} = D^{(a)}(i, m_i) - D^{(a)}(1, i)$.
  Find k and j resulting most negative trade-off value,
  $k \leftarrow \min_i(t^{(a)}), j \leftarrow m_k$.
  do
   Add new nodes $N_{new} \leftarrow k + j$ if not exist in current trees
   if weight of merging subtree of $N_{new} < C_a$
    Add N_{new} to corresponding tree i
    Update cumulative weight of corresponding tree i, $C_i^{(a)}$.
    Declare N_{new} is accepted.
   else
    update $D^{(a)}(k, j) \leftarrow \infty$
    Search for next min cost neighbor for k,
    $m_k \leftarrow \min_{m_k} D^{(a)}(k, m_k)$.
    Recompute trade-off for k, $t_k^{(a)} = D^{(a)}(k, m_k) - D^{(a)}(1, k)$.
    Declare N_{new} rejected.
  until (N_{new} is accepted) or (all $t_i^{(a)} \geq 0$)
end while
Select a tree results maximum energy efficiency.

from the prioritized set $\mathcal{N}_p^{(a)}$ into Ψ while ensuring the battery deadline in Eq. (23) for all nodes are still met. To this end, we sort the nodes in $\mathcal{N}_p^{(a)}$ in a descending order of residual lifetimes and denote the sorted sequence as Ω. We insert these nodes starting from the first node Ω_1. Let A_i denote the arrival time of the charging vehicle at the i-th node in the shortest path Ψ, $i = \{1, 2, \ldots, n_f^{(a)}\}$.

To insert the j-th node Ω_j from Ω into Ψ, we first find position m_t in Ψ such that $A_{m_t} \leq l_{\Omega_j}$ and $A_{m_t+1} > l_{\Omega_j}$ where l_{Ω_j} is Ω_j's lifetime. We call m_t the *temporary maximum position* to insert Ω_j. It indicates the maximum number of nodes in Ψ that can be served before node Ω_j depletes its battery. To accommodate the remaining $|\Omega| - j$ nodes, we need to find a position such that even all the remaining nodes are inserted before Ω_j, Ω_j can still meet its battery deadline. We find the maximum position m such that $A_m \leq A_{m_t} - \sum_{i=j+1}^{n_p^a} t_i$ and $A_{m+1} > A_{m_t} - \sum_{i=j+1}^{n_p^a} t_i$, where t_i is the recharge time of Ω_j. Now, the maximum position m represents the rightmost position Ω_j can be inserted if all remaining nodes are later inserted before Ω_j.

For each of the m possible positions that Ω_j can be inserted, a total traveling cost is computed and the one that minimizes the traveling cost is selected as the final insertion position for Ω_j. Then we obtain a new sequence Ψ and remove Ω_j from Ω. The iteration continues until we exhaust Ω or an infeasible situation is encountered. Table III shows the pseudo-code of the insertion algorithm.

We briefly illustrate how the insertion algorithm works in Fig. 5. We consider two nodes Ω_1, Ω_2 with lifetime 104 mins and 90 mins that need to be inserted into a feasible recharge sequence. We find the position k to insert Ω_1 is between node 6 and 7 since $A_6 < l_{\Omega_1} < A_7$. To ensure Ω_1 can survive when Ω_2 is later inserted before Ω_1, k' can only be between node 3 and 4 (since $A_3 < A_6 - T_{\Omega_1} < A_4$). Then we search all the 4 possible locations (before node 1, 2, 3, 4) and find that the position before node 3 minimizes the traveling cost. Thus Ω_1 is inserted between node 2 and 3. We repeat the procedure for Ω_2. Since it is the last node, we can directly calculate the rightmost insertion position k' and find the minimum cost among

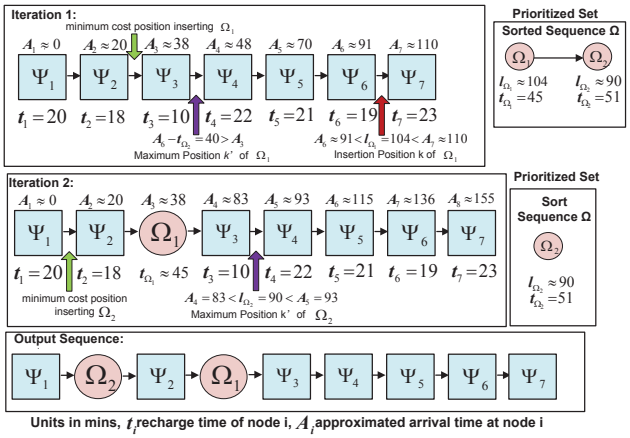


Fig. 5. Illustration of insertion algorithm.

TABLE III
INSERTION ALGORITHM

input: CMST $\mathcal{N}_r^{(a)}$, lifetime l_i and recharge time t_i, $i \in \mathcal{N}_r^{(a)}$, distance matrix $D^{(a)}$, feasible set $\mathcal{N}_f^{(a)}$ satisfying Proposition 2.
output: resultant recharge sequence Ψ.
Compute shortest path in the feasible set, $\Psi \leftarrow \text{TSP}(\mathcal{N}_f^{(a)})$
Sort $\mathcal{N}_p^{(a)}$ in a descending order of lifetime as Ω
Initialize $i \leftarrow 1$, last step node position $k \leftarrow \infty$.
while $\Omega \neq \emptyset$
  Find temporary max position m_t in Ψ such that
  $A_{m_t} \leq l_{\Omega_i}$ and $A_{m_t+1} > l_{\Omega_i}$
  Find the max insertion position m such that
  $A_m \leq A_{m_t} - \sum_{k=i+1}^{n_r^p} t_k$ and $A_{m+1} > A_{m_t} - \sum_{k=i+1}^{n_r^p} t_k$.
  if Cannot find $m \geq 0$. **break**,return infeasible and report.**end if**
  Set minimum cost $c_{min} \leftarrow \infty$.
  for x from 0 to m
   Insert node Ω_i into Ψ, get temporary sequence Ψ_t
   Calculate cost $c \leftarrow \sum_{j=1}^{|\Psi_t|-1} D^a(j, j+1)$.
   if $c < c_{min}$, $\Psi \leftarrow \Psi_t$, $c_{min} \leftarrow c$, $k \leftarrow x$. **end if**
  end for
  $i \leftarrow i + 1$, update $\Omega \leftarrow \Omega - i$
end while
Return recharge sequence Ψ, minimum cost c_{min}.

possible inserting positions.

Remarks: Due to the randomness in sensors' lifetimes, it is very difficult to derive a theoretic performance bound of the algorithm. However, we have conducted simulations in Section VII-A and find our algorithm has about 1.06 approximation ratio to the optimal solution.

D. Complexity Analysis

We now analyze the complexity of our algorithms. The complexity of the greedy algorithm is $\mathcal{O}(n)$ because it only selects the maximum profitable node at each step. For the adaptive algorithm, the base station has abundant resources and it performs the k-means algorithm. So we focus on the computing burdens on charging vehicles for calculating CMST and route improvements. In the worst case, there is only one charging vehicle to recharge n nodes. For the extended EW algorithm, finding the minimum trade-off value requires n^2+2n iterations at the outer loop. In the inner loop, the worst case is that for a node with the minimum trade-off value, every minimum-cost neighbor is rejected due to capacity violations. So n iterations are required. In sum, its time complexity is $\mathcal{O}(n^3)$.

For the route improvement algorithm, running a TSP algorithm requires $\mathcal{O}(n^2)$ time. Sorting nodes' lifetimes requires $\mathcal{O}(n \log n)$ time. Then, insertion requires $\mathcal{O}(n^2)$ time. Hence, the total time complexity of route improvement algorithm is $\mathcal{O}(n^2)$ and the

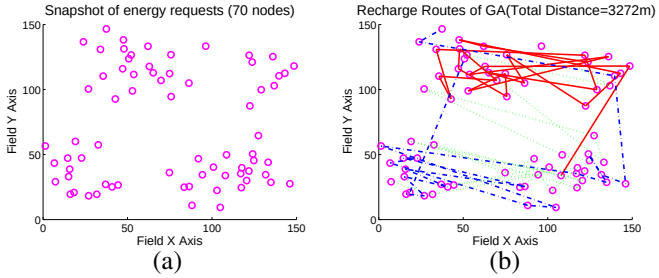


Fig. 6. An example of the Greedy Algorithm (a) a snapshot of recharge request. (b) recharge routes from the Greedy Algorithm.

adaptive algorithm takes $\mathcal{O}(n^3)$ time. Note that although the proposed algorithms are centralized, they run on the charging vehicles that usually have much higher computation and energy resources than common sensor nodes. It is not difficult for them to handle computations for large networks.

E. An Example of Algorithms

To illustrate operations of the algorithms, we show an example in Fig. 6-Fig. 7. A snapshot of 70 recharge requests from sensors during the operation is presented in Fig. 6(a) when three charging vehicles cooperate to recharge these nodes. Fig. 6(b) demonstrates the recharge routes using the Greedy Algorithm with a total distance of 3272 m. We can see the charging vehicles travel long distances to take care of energy requests in the field, which matches our analysis in Section VI-B. Fig. 7(a) shows an adaptive network partitioning of the recharge requests into three regions. Then the charging vehicles compute the CMST in parallel fashion in Fig. 7(b). Note that two trees are generated for charging vehicle 1 due to limited recharge capacity. The tree with higher ratio between energy demands and sum of edge costs is chosen first. The uncovered nodes will be recharged in the next round after the charging vehicle has replenished its own battery at the base station. Next, each charging vehicle calculates an improved recharge route on the selected tree shown in Fig. 7(c). Charging vehicle 1 has to return to base station for battery replacement before recharging nodes on the second tree (edges shown as dashed line). In contrast to the Greedy Algorithm, charging vehicles only travel a distance of 993 m which suggests great potentials of the Adaptive Algorithm to reduce system cost.

VII. Performance Evaluations

We have developed a discrete event-driven simulator using POSIX multi-thread programming in C language. In our simulator, packet transmissions between nodes are modeled by inter-thread communications and each vehicle also calculates the recharge decisions by exchanging information. To model WRSNs with high accuracy, the simulator takes real parameters such as battery recharge times.

A number of $N = 500$ sensor nodes are uniformly randomly deployed over a square sensing field with side length $L = 160$ m. All sensors transmit at the same power level with fixed transmission range $d_r = 15$ m. The choice of maximum cluster hop-count k will have a direct impact on energy consumption and data gathering latency. On one hand, a large k would result in large intra-cluster energy consumptions due to more traffic relays, especially on anchor points which aggregate all the packets. This would potentially increase the load on charging vehicles. On the other hand, a small k will generate more clusters. To cover all the nodes, the data collection tour would be elongated and cause higher latency. Through trials we find that when $k = 3$, $c \approx 5$ clusters are needed to cover the entire field, and the intra-cluster energy consumptions are not too large. Thus we set $k = 3$. Dijkstra's shortest path algorithm is used to route packets from sensors to their corresponding anchor points at an average rate of $\lambda = 3$ pkt/min and 30 bits per packet

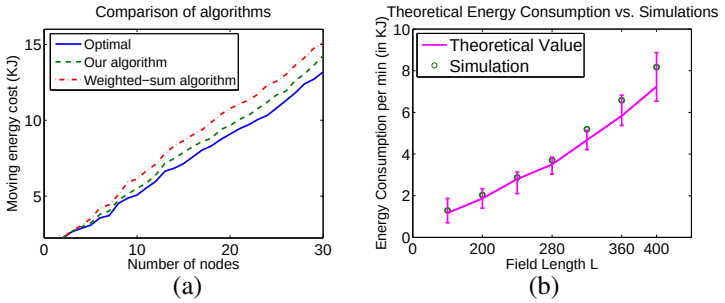


Fig. 8. Evaluation of algorithms and validation of theoretical model (a) comparison of different algorithms (b) validation of energy consumption model.

following a Poisson process. Each time slot is 1 min. The bit rate is 250 Kbps. Since a higher initial energy takes longer time for the network to achieve equilibrium, we set all sensors to start from 50% battery initially to make the network enter equilibrium faster. The charging vehicles collect energy information every 12 hours and each time it finishes fulfilling all the energy requests.

Sensor nodes have adaptive recharge thresholds regarding their communication hop counts to anchor points following Eq. (17). Given $\tau_1 = 0.75$, we can calculate $\tau_2, \tau_3 = 0.57, 0.22$, respectively. The battery's recharge time is modeled from [23]. We assume charging vehicles are electric-powered vehicles carrying computing, communication modules and high density battery packs (e.g., 12A, 5V standard ones). The vehicle can weight tens of pounds and we assume it is 20 lbs. Using the method in [27], we estimate that a vehicle consumes energy at a rate of 5.59 J/m. To evaluate how the number of charging vehicles affects system performance as well as validate theoretical results in Proposition 1, we vary the number of charging vehicles m from 1 to 5 and set the simulation time to 4 months.

A. Evaluation of Algorithm and Energy Consumption Model

We first evaluate the performance of the adaptive algorithm by comparing with the optimal solution and *weighted-sum algorithm* proposed in [12]. The weighted-sum algorithm finds the shortest recharge sequence based on traveling time and residual lifetime of sensor nodes through a weighted parameter. It tries different weighted parameters and chooses the best solution among all the trials. Both the weighted-sum and adaptive algorithms aim to capture the battery deadline constraints.

Due to the NP-hardness of our problem, it is very difficult to obtain optimal solutions using brute force for large networks. To provide a baseline for comparison, we have managed to obtain optimal solutions for networks up to 30 nodes by pruning solutions that lead to infeasibility or suboptimality. We set the residual energy of sensor nodes uniformly randomly distributed from zero to 20% and compare different approaches that form recharge routes through all the nodes. The simulation results are averaged over 100 datasets. Fig. 8(a) shows the moving energy consumption of charging vehicles. We can see that for a small network size (1-5 nodes), the gaps between our adaptive algorithm and the optimal solution is small. This is because the number of different possible schedules is small. Our algorithm may find the optimal schedule, or one very close. What is interesting is that the ratio remains almost the same as we increase the number of nodes. The maximum ratio of 1.10 appears when the number of nodes is 14. On average, the ratio is 1.065 to the optimal solution, which offers a good approximation. This shows our algorithm can still find schedules very close to optimal even when the search space has grown dramatically. For the weighted-sum algorithm, the maximum ratio is 1.22 when we have 8 nodes, and the average approximation ratio is 1.16. The results

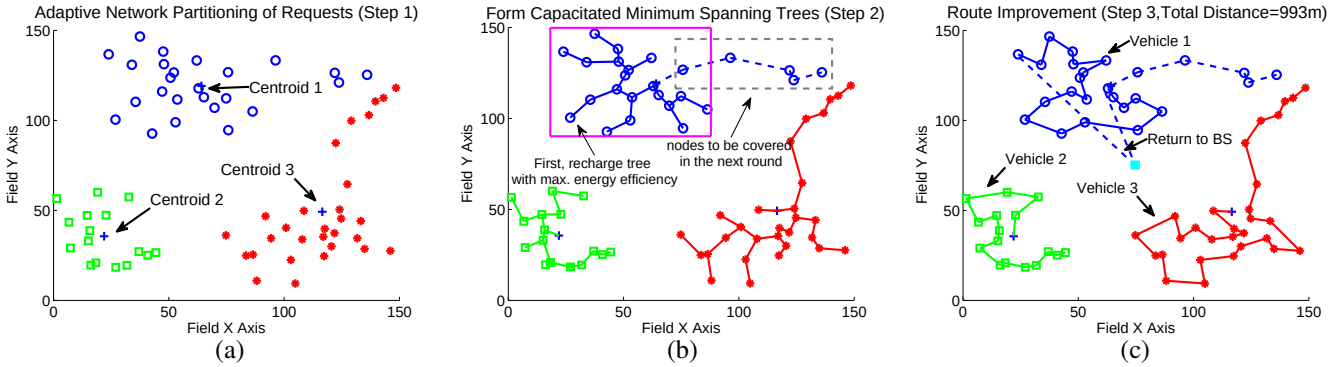


Fig. 7. An example of the Adaptive Algorithm (a) adaptive network partitioning regarding recharge request. (b) establish CMST (c) improve recharge route.

indicate that the adaptive algorithm saves an additional 8% energy cost compared to the weighted-sum algorithm. Besides, the selection of weight parameter in [12] may not be easy in real applications. The adaptive algorithm utilizes an existing solution from the TSP problem without the complexity to examine various weight values.

We also evaluate the correctness and accuracy of the energy consumption model shown in Fig. 8(b). To examine our model over different network field sizes, we first set $N = 500$, $L = 160$m (node density $\rho = 0.019$ nodes/m^2) and increase L from $160 - 400$ m while keeping node density the same. The theoretical results show the average energy consumptions with variations along the curve. That is, if we use the lower bound of Eq. (1) to calculate the number of anchor points, we have a lower limit for the energy consumption. On the other hand, if we count anchor points according to actual layouts governed by Eqs. (3), (4) and (5), an upper bound on energy consumption is derived (it overestimates partial clusters on the boundaries). It is observed that our energy consumption model can achieve very high accuracy (falls within theoretical variations). For $L = 160 - 280$, the simulation results almost match our theoretical model and for $L = 320 - 400$m, the simulation results are within 15% of the average theoretical numbers. The inaccuracies are due to an increasing number of clusters on the field boundaries, which are not complete circles causing overestimates. Next, we will validate the entire theoretical model on the minimum number of charging vehicles.

B. Evaluation of Network Performance

In this subsection, we evaluate the performance of proposed algorithms in terms of the number of nonfunctional nodes, energy consumption vs. replenishment, recharge fairness, duration of nonfunctional nodes, data collection latency and operating energy cost.

1) Nonfunctional Nodes: First, we examine the evolution of the number of nonfunctional nodes. When a sensor node depletes its battery energy, it becomes nonfunctional until recharged. Fig. 9 presents the results of nonfunctional nodes by proposed algorithms.

For the Greedy algorithm (GA), when $m = 1$, the number of nonfunctional nodes surges dramatically around 18 days to over 80% until it slowly decreases and stabilizes at 55% around 37 days. Similar phenomenons are observed for $m = 2, 3$. This is because the charging vehicles favors nodes closer to the anchor points with more recharge profits. Thus they do not serve nodes in the outmost corona of clusters fast enough after their requests. Charging vehicles only cover them when their batteries nearly deplete. By then, their recharge capacity ($m = 2$) is temporarily exceeded, which causes the big spike. Although $m = 1 - 3$ charging vehicles can gradually resolve most nonfunctional nodes, it is observed that there is persistently more than 50%, 20% and 10% nonfunctional nodes for $m = 1, 2, 3$, respectively. In contrast, the Adaptive Algorithm (AA) provides more stability. When $m = 2, 3$, there is no such

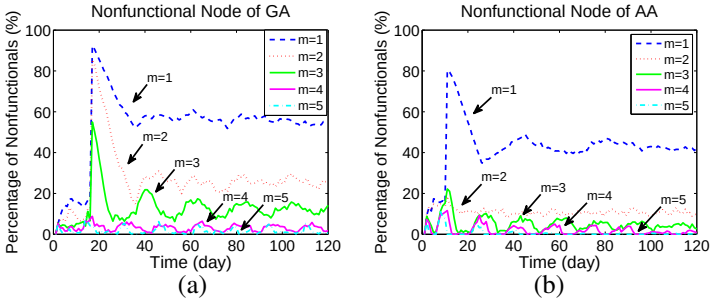


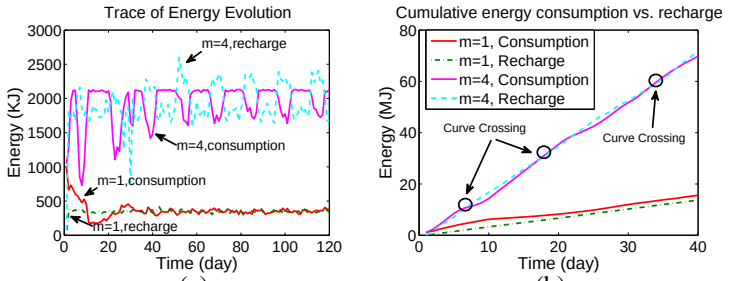
Fig. 9. Evolution of nonfunctional nodes. (a) GA. (b) AA.

huge spike. For $m \geq 3$, nonfunctional nodes are within 10% at network equilibrium. This is because AA captures the sensor battery deadlines. When $m = 5$, AA can reduce the nonfunctional nodes to zero.

We observe that $m = 5$ is likely to be a threshold since 4 charging vehicles still result in sporadic 5% nonfunctional nodes. From *Proposition 1*, after plugging in the experimental parameters, we obtain $m = \lceil 4.72 \rceil = 5$. Thus $m = 4$ can barely satisfy the energy neutral condition. This calculation matches our observations in Fig. 9(b), validating the correctness of our theoretical results.

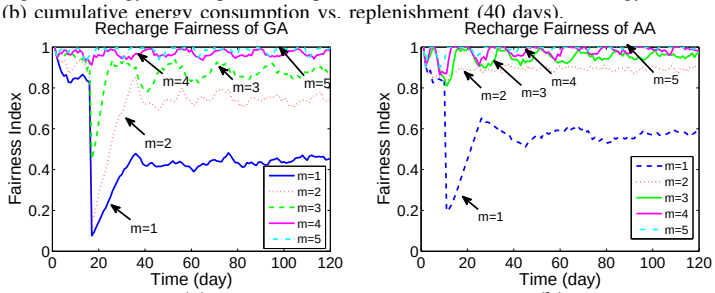
2) Energy Consumption vs. Recharge: In this subsection, we demonstrate the evolution of energy consumption vs. replenishment. Since GA and AA have similar curve shapes, we illustrate the energy changes of AA only. In Fig. 10(a), we trace the evolution of consumed and replenished energy when $m = 1, 4$. For $m = 1$, it is definitely not enough to sustain network operations. Thus nodes continuously deplete battery and no longer consume energy, which causes the drop in energy consumption at the very beginning. Since the recharge capacity of one vehicle puts an upper limit on the energy consumption, the two curves reach an equilibrium and converge after 30 days. For $m = 4$, about 4 times the energy is replenished compared to $m = 1$, thus the large gap in between. We also observe that when there is a drop in energy consumption, the energy replenishment correspondingly jumps up, which represents four vehicles acting in response to to battery depletions.

To illustrate energy balance in the network, we also show the cumulative energy evolution in Fig. 10(b). For clarity and better observing the gaps and intersections between curves, we plot 40 days' simulation time. If the energy replenishment curve is above the consumption curve, more energy has been refilled into the network than consumed, and vice versa. For $m = 1$, the energy consumption curve is above the energy replenishment curve. A larger gap is observed at the first 10 days, indicating energy replenishment can barely keep up with consumptions. In contrast, with $m = 4$, the energy consumption curve stays above replenishment until the two curves first cross each other around 6 days. This is because from the very beginning, more energy is consumed than replenished. Around 6 days, a few nodes have depleted energy and stopped consuming



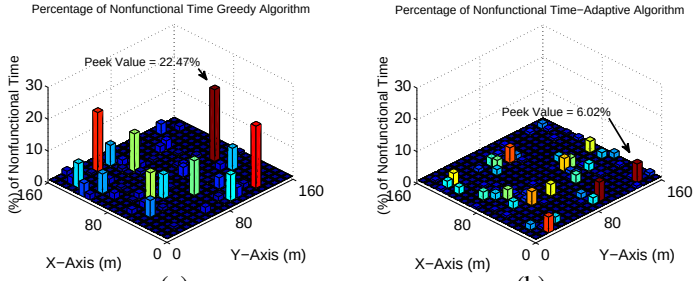
(a) (b)

Fig. 10. Energy consumption vs. replenishment of AA. (a) trace of energy evolution. (b) cumulative energy consumption vs. replenishment (40 days).



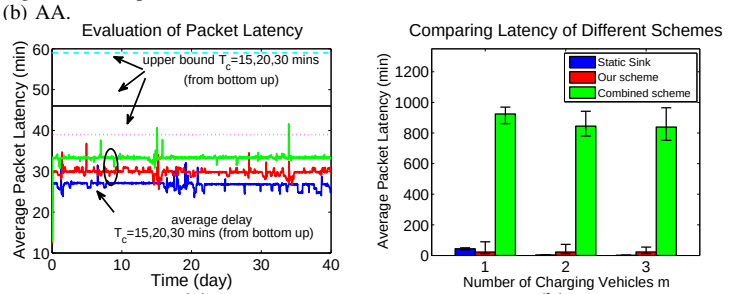
(a) (b)

Fig. 11. Comparison of recharge fairness. (a) GA. (b) AA.



(a) (b)

Fig. 12. Comparison of durations for nonfunctional nodes when $m = 4$. (a) GA. (b) AA.



(a) (b)

Fig. 13. Evaluation of data collection latency (a) Latency using different T_c vs. upper bound. (b) Comparison of latency between different data collection schemes.

more, which brings down the consumption curve. The replenishment curve stays above the consumption curve until the next crossing around 20 days. Therefore, the evolution of network energy also validates $m = 4$ is a threshold case since sporadic battery depletions are observed.

3) Recharge Fairness: Recharge fairness indicates whether charging vehicles recharge nodes commensurate to their workloads. Those having more workload (e.g., nodes near the base station) should be recharged more frequently. This is reflected from the functional time of sensor nodes. To quantify recharge fairness, we leverage the fairness index from [39],

$$F = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} (x_i)^2}, \tag{29}$$

in which x_i is a normalized indicator whether a node is functional in a time slot. x_i equals $1/N_s$ if i is functional in a time slot, otherwise, it is zero. The fairness index F ranges from 0 (worst case if all nodes are nonfunctional) to 1 (best case if all the nodes are functional). In Fig.11(a), when nodes in the outmost ring become nonfunctional, the fairness of GA algorithm severely degrades as vehicles only recharge nodes with maximum profits. We can see from Fig.11(b) that AA can distribute energy resources fairly among the nodes especially when $m = 4, 5$ ($F = 1$).

4) Nodes' Nonfunctional Periods: Fig. 12 plots the percentage of nonfunctional durations of nodes as a function of their locations. Using GA, nodes near the anchor points have a maximum of 22.47% time in nonfunctional states whereas AA is only 6.02%. Further, AA spreads nonfunctional durations across the field while the spikes of GA are highly concentrated around anchor point locations. This is because nodes close to anchor points consume energy faster and are more prone to become nonfunctional. GA considers profit only and has no measure for battery deadlines. In contrast, AA considers both profit and battery deadlines. Therefore, the duration of nonfunctional nodes with AA is significantly less than that of GA.

5) Data Collection Latency: Data collection latency mainly depends on two variables: dispatching time interval T_c and availability of routing paths. The former is a system parameter determining how often to dispatch the data gathering vehicle; the later relies on the number and locations of nonfunctional nodes. To transmit packets to anchor points timely, all nodes should be functional on a routing path. We assume shortest routing paths by Dijsktra's algorithm are used. If a node depletes energy and there is no alternate route available, all pending messages are buffered at senders until the path is restored.

To see the fluctuations of curves more clearly, we trace the evolution of data collection latency for the first 40 days in Fig. 13(a). We vary $T_c = 15, 20, 30$ mins, resulting in average packet latencies of 27,30,34 mins respectively. The small spikes are caused by temporary unavailability of routing paths when packets are buffered for longer time. We have also plotted corresponding data collection upper bounds calculated in *Lemma 2* and we observe that the actual packet latencies are well within these bounds. It is interesting to see when $T_c = 15$ mins, the latency is 27 mins whereas when $T_c = 30$ mins, the latency only increases slightly to 34 mins. This is because data transmission time and vehicle's moving time dominate when $T_c = 15$ mins. This indicates that sending out the data gathering vehicle too frequently may not help reduce packet latency too much compared to the extra operating costs incurred.

In addition, we have also compared packet latency between different data collection schemes. A static data sink is used in [14], [17] to gather all the packets and we denote it as "Static". A combination of data sink and wireless charging on a single vehicle is proposed in [15] and we denote it as "Combined". Fig. 13(a) compares the average packet latency when we increase the charging vehicles from 1 to 3. First, we can see both the static and our schemes have about two orders of magnitude less latency than the combined scheme. The large latency of the combined scheme is caused by the inevitable gap between battery recharge time and data transmission time. The delivery of gathered data has to wait for at least 10 hours until the charging vehicle returns to the base station for battery replacement. On the contrary, our scheme employs a dedicated vehicle without any waiting for recharge. Second, although the static scheme is expected to yield less latency than our scheme (when $m = 2, 3$), it has a higher latency when $m = 1$. Since using a static sink results in more traffic relays, thus higher energy consumption. When there are not enough charging vehicles, nonfunctional nodes lead to unavailability of routing paths and longer latencies.

6) Operating Energy Cost: In this subsection, we evaluate the traveling energy cost of charging vehicles. Fig. 14(a) compares the average traveling cost per vehicle for GA and AA. When $m = 1-3$,

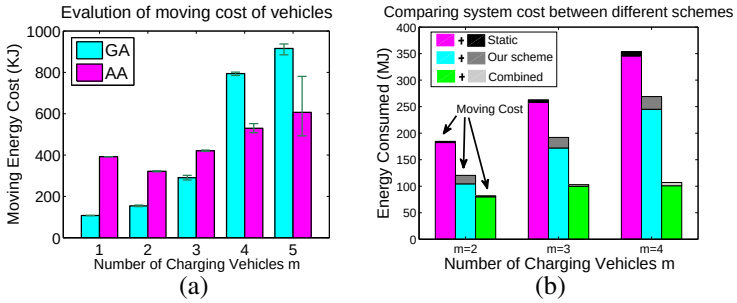


Fig. 14. Evaluation of operating energy cost. (a) Comparing charging vehicle's moving cost between GA and AA. (b) Comparing total system cost between different data collection schemes.

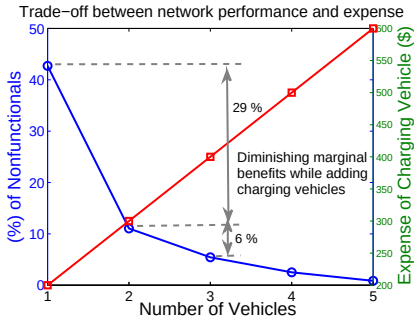


Fig. 15. Trade-off between network performance and expense.

more energy cost is observed with AA. This is because the AA takes care of nodes in the outmost corona preemptively before they deplete energy, thus more energy is used in traveling. With GA, charging vehicles travel to the outmost corona only when recharge profits there are larger, but by then those nodes nearly deplete energy and many become nonfunctional. Although GA has lower traveling cost when $m = 1-3$, the network performance deteriorates greatly. When $m = 4-5$, we can partition the network into more regions with smaller sizes so the movements of charging vehicles can be confined in smaller regions. This brings down the movement energy for charging vehicles. However, as GA does not partition the network, long distance travels are inevitable. So AA incurs less energy with more charging vehicles.

Fig. 14(b) shows the total system cost on vehicles for different data collection schemes. For fair comparison, we set the communication hop count $k = 3$ in both our scheme and the combined scheme in [15]. The main body of the bar charts are energy used for recharging sensor nodes and the dark portion on top represents the total moving energy cost on vehicles. First, we can see the static scheme used in [17] consumes most energy since multi-hop forwarding to the base station requires more hops of traffic relays. Although introducing a dedicated data gathering vehicle increases the moving cost, the total system cost is still 30% less than the static scheme. This is because we have smaller clusters and thus less energy for traffic relay on intermediate nodes. The combined scheme seems to have the least system cost. However, it has prohibitive network latency as illustrated in Fig. 13(b). Further, since the data collection hop count $k = 3$, it is possible that some nodes are not covered in simulation time. So their packets have to be buffered until the vehicle moves into multi-hop communication range. It lowers the energy consumption at the cost of dramatically exacerbating packet latency.

7) Trade-off between Network Performance and Expense: Finally, we evaluate the trade-off between network performance and monetary costs of the charging vehicles. We assume one vehicle may not cost too much (e.g., $100) when manufactured at large scale. Fig. 15 shows the average percentage of nonfunctional nodes versus the total costs of vehicles. Initially there are one charging and one data collecting vehicles, resulting in nearly 42% nonfunctional nodes. Adding one more charging vehicle reduces this number to 12%. As we keep adding charging vehicles, the marginal benefits decrease whereas the expense grows linearly. This shows that when the number of nonfunctional nodes is already very small (e.g. below 10%), adding more charging vehicles may not be cost-effective. Therefore, considering such trade-offs, a good strategy is to select a minimum number of charging vehicles that can maintain very low levels (e.g. around 5%) of nonfunctional nodes.

VIII. Discussions

We discuss a couple interesting issues worth future study. For large scale networks or nodes having different packet delay requirements, multiple data gathering vehicles and base stations might be needed. The first question is how to dispatch and coordinate multiple data gathering vehicles such that the packet latency deadlines of different clusters are satisfied. The problem is analogous to the charging problem in Section VI-C3 since packet delivery latency is similar to battery deadlines. However, a cluster might be visited by multiple data gathering vehicles within a short time period [21], [22]. Second, where to place multiple base stations so as to minimize the traveling distance of data gathering vehicles through anchor points. This is a *location-routing problem* which is NP-hard [40]. Considering these two problems in an integrated solution is even more difficult given the dependency between them. We plan to study these problems in the future.

IX. Conclusions

In this paper, we consider several important factors overlooked by previous WRSN studies, including the charging vehicle's energy consumption, capacity limits, energy efficiency and data latency. We first propose a low latency mobile data gathering scheme that can collect packets from all nodes and provide theoretical results on latency. Then we establish a mathematical model to calculate the minimum number of charging vehicles needed, nodes' lifetimes and adaptive recharge thresholds. We formulate recharge optimization problem into a Profitable Traveling Salesmen Problem with Capacity and Battery Deadline constraints, which is NP-hard. We propose two low complexity algorithms. The greedy algorithm maximizes the recharge profit in each step. A three-step adaptive algorithm systematically captures the recharge capacity and nodes' battery deadline constraints while minimizing traveling costs. We evaluate and compare the proposed algorithms by extensive simulations. They show that the adaptive algorithm can provide better stability by reducing the number of nonfunctional nodes and their nonfunctional duration lengths. We also validate the theoretical results through simulations. The comparison with other schemes show that the adaptive algorithm achieves both low latency and high energy efficiency.

X. Acknowledgments

The work in this paper was supported in part by the grant from US National Science Foundation under grant number ECCS-1307576.

References

[1] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances," *Science*, vol. 317, pp. 83, 2007.
[2] PowerCast Corp, "http://www.powercastco.com."
[3] X. Wu, G. Chen and S. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," *IEEE TPDS*, vol.19, no.5, 2008.
[4] B. Tong, Z. Li, G. Wang and W. Zhang, "How wireless power charging technology affects sensor network deployment and routing," *IEEE ICDCS*, 2010.
[5] Y. Peng, Z. Li, W. Zhang and D. Qiao, "Prolonging sensor network lifetime through wireless charging," *IEEE RTSS*, 2010.
[6] Z. Li, Y. Peng, W. Zhang, and D. Qiao, "J-RoC: a joint routing and charging scheme to prolong sensor network lifetime," *IEEE ICNP*, 2011.

[7] S. He, J. Chen, F. Jiang, D. Yau, G. Xing and Y. Sun,"Energy provisioning in wireless rechargeable sensor networks," *IEEE Trans. Mobile Computing*, vol. 12, no. 10, pp. 1931-1942, Oct. 2013.

[8] Y. Shu, P. Cheng, Y. Gu, J. Chen, T. He," TOC: Localizing wireless rechargeable sensors with time of charge," *ACM TOSN*, vol. 11, no. 3, pp. 1-22, 2015.

[9] H. Dai, Y. Liu, G. Chen, X. Wu, T. He, " SCAPE: Safe charging with adjustable power", *IEEE ICDCS*, 2014.

[10] S. Nikoletseas, R. Theofanis and R. Christoforos, "Low radiation efficient wireless energy transfer in wireless distributed systems," *IEEE ICDCS*, 2015.

[11] C. Angelopoulos, S. Nikoletseas, T. Raptis, C. Raptopoulos, F. Vasilakis,"Efficient energy management in wireless rechargeable sensor networks," *IEEE MSWiM*, 2012.

[12] C. Wang, J. Li, F. Ye and Y. Yang, "Multi-Vehicle coordination for wireless energy replenishment in sensor networks," *IEEE IPDPS*, 2012.

[13] Y. Yang and C. Wang, "Wireless Rechargeable Sensor Networks",*Springer*, 2015.

[14] Y. Shi, L. Xie, T. Hou and H. Sherali, "On renewable sensor networks with wireless energy transfer," *IEEE INFOCOM*, 2011.

[15] M. Zhao, J. Li and Y. Yang, "A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," *IEEE Trans. Mobile Computing*, vol. 13, no. 12, 2014, pp. 2689-2705.

[16] L. Xie, Y. Shi, Y. T. Hou, W. Lou, H. Sherail and S. Midkiff, "Bundling mobile base station and wireless energy transfer: Modeling and optimization," *IEEE Infocom*, 2013.

[17] C Wang, J Li, F Ye, Y. Yang, "Recharging Schedules for Wireless Sensor Networks with Vehicle Movement Costs and Capacity Constraints",*IEEE SECON*, 2014.

[18] Online: "http://www.afar.net/tutorials/fcc-rules".

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

[20] W. Wang, V. Srinivasan and K. Chua, "Extending the lifetime of wireless sensor networks through mobile relays," *IEEE/ACM Trans. Networking*, vol. 16, no. 5, pp. 1108-1120, 2008.

[21] A. Somasundara, A. Ramamoorthy and M. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," *IEEE RTSS*, 2004.

[22] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner and C. Lee, "Paritioning based mobile element scheduling in wireless sensor networks," *IEEE SECON*, 2005.

[23] Panasonic Ni-MH battery handbook, "http://www2.renovaar.ee/userfiles/Panasonic_Ni-MH_Handbook.pdf".

[24] R. Kershner, "The number of circles covering a set," *American Journal of Mathematics*, vol. 61, pp. 665-671, 1939.

[25] V. Rai and R. N. Mahapartra, "Lifetime modeling of a sensor network,"*IEEE DATE*, 2005.

[26] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *IEEE HICSS*, 2000.

[27] Battery Calculator, "http://www.evsource.com/battery_calculator.php."

[28] S. Ross, *A First Course in Probability*, 8th Ed, Prentice Hall, 2009.

[29] D. Feillet, P. Dejax and M. Gendreau, "Traveling salesman problems with profits," *Transportation Science*, vol. 39, no. 2, 2005.

[30] B. Gavish, "A note on the formulation of the m-salesman traveling salesman problem," *Management Science,* 1976.

[31] M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh and F. Soumis, "Vehicle routing with time windows: optimization and approximation," Elsevier Science Publisher, 1988.

[32] M.W.P. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operation Research*, pp. 285-305, 1985.

[33] N. Bansal, A. Blum, S. Chawla and A. Meyerson, "Approximation algorithms for Deadline-TSP and Vehicle Routing with Time Windows," *ACM STOC*, 2004.

[34] B. Chandran and S. Raghavan, "Modeling and solving the capacitated vehicle routing problem on trees," *The Vehicle Routing Problem: Latest Advances and New Challenges*, pp 239-261, 2008.

[35] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proc. of 5th Berkeley Symposium on Math. Statistics and Probability*, 1967, pp. 281-97.

[36] L.R. Esau and K.C. Williams, "On teleprocessing system design: part II-a method for approximating the optimal network," *IBM Syst Journal*, vol. 5, pp. 142-147, 1966.

[37] T.H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, MIT Press, 2001.

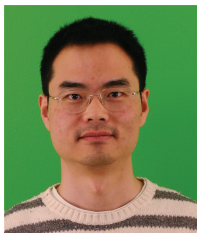
[38] P. Jaillet, "Probabilistic traveling salesman problem," Ph.D. Dissertation, MIT, 1985.

[39] R. Jain, D. M. Chiu, W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," *DEC Research Report TR-301*.

[40] G. Nagy and S. Salhi, "Location-routing: issues, models and methods," *European Journal of Operation Research*, no. 177, pp. 649-672, 2007.

Cong Wang received the BEng degree in Information Engineering from the Chinese University of Hong Kong and M.S. degree in Electrical Engineering from Columbia University, New York. He is currently working towards the PhD degree at the Department of Electrical and Computer Engineering, Stony Brook University, New York. His research interests include wireless sensor networks, performance evaluation of network protocols and algorithms.

Ji Li received the B.S. degree in Electrical Engineering from Harbin Engineering University, Harbin, China, and the M.S. degree in Electrical Engineering from Zhejiang University, Hangzhou, China. He is currently a PhD student in the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, New York. His research interests include wireless sensor networks and embedded systems.



Fan Ye received his B.E. and M.S. degrees in Automation and Computer Science from Tsinghua University, Beijing, China, and Ph.D. in Computer Science from UCLA. He then joined IBM T. J. Watson Research as a Research Staff Member, working on stream processing systems, cloud messaging and mobile computing. He is currently an assistant professor in the Department of Electrical and Computer Engineering at Stony Brook University. He holds over a dozen US/international patents and patent applications. His research interests include mobile computing, mobile cloud, wireless networks, sensor networks and their applications.

Yuanyuan Yang received the BEng and MS degrees in computer science and engineering from Tsinghua University, Beijing, China, and the MSE and PhD degrees in computer science from Johns Hopkins University, Baltimore, Maryland. She is a professor of computer engineering and computer science at Stony Brook University, New York, and the director of Communications and Devices Division at New York State Center of Excellence in Wireless and Information Technology (CEWIT). Her research interests include wireless networks, data center networks, optical networks and high-speed networks. She has published more than 270 papers in major journals and refereed conference proceedings and holds seven US patents in these areas. She is currently the Associate Editor-in-Chief for the IEEE Transactions on Computers and an Associate Editor for the Journal of Parallel and Distributed Computing. She has served as an Associate Editor for the IEEE Transactions on Computers and IEEE Transactions on Parallel and Distributed Systems. She has served as a general chair, program chair, or vice chair for several major conferences and a program committee member for numerous conferences. She is an IEEE Fellow.