2010 31st IEEE Real-Time Systems Symposium

Prolonging Sensor Network Lifetime Through Wireless Charging*

Yang Peng, Zi Li, Wensheng Zhang, and Daji Qiao
Iowa State University, Ames, IA, USA
Email: {yangpeng,zili,wzhang,daji}@iastate.edu

Abstract

The emerging wireless charging technology is a promising alternative to address the power constraint problem in sensor networks. Comparing to existing approaches, this technology can replenish energy in a more controllable manner and does not require accurate location of or physical alignment to sensor nodes. However, little work has been reported on designing and implementing a wireless charging system for sensor networks. In this paper, we design such a system, build a proof-of-concept prototype, conduct experiments on the prototype to evaluate its feasibility and performance in small-scale networks, and conduct extensive simulations to study its performance in large-scale networks. Experimental and simulation results demonstrate that the proposed system can utilize the wireless charging technology effectively to prolong the network lifetime through delivering energy by a robot to where it is needed. The effects of various configuration and design parameters have also been studied, which may serve as useful guidelines in actual deployment of the proposed system in practice.

1 Introduction

Wireless sensor nodes are powered by small batteries, and the limited energy supply has constrained the lifetime of a sensor network. This has been a long-lasting, fundamental problem faced by sensor networks that are designed for long-term operation. Energy conservation [2, 8], environmental energy harvesting [3, 5, 6, 11], incremental deployment, and battery replacement [17, 19] approaches have been proposed to address the problem. However, energy conservation schemes can only slow down energy consumption but not compensate energy depletion. Harvesting environmental energy, such as solar [3, 6], wind [11], vibration [5], is subject to their availability which is often uncontrollable by people. The incremental deployment approach may not be environmentally friendly because deserted nodes can pollute the environment. The battery or node replacement approach is applicable only for scenarios that sensor nodes are accessible by people or sophisticated robots that can locate and physically touch the sensor nodes.

*This work is supported partly by the NSF under Grant CNS-0831874.

As the wireless charging technology is being commercialized [12], it has become a promising alternative to address the energy constraint problem in sensor networks. Different from energy harvesting technologies, the wireless charging technology, together with more and more mature and inexpensive mobile robots, will make controllable power replenishment possible, with which power can be replenished proactively to meet application requirements rather than passively adapted to the availability of environmental resources. Comparing with sensor node or battery replacement approaches, the wireless charging technology allows a mobile charger to transfer energy to sensor nodes wirelessly without requiring accurate localization of sensor nodes or strict alignment between the charger and nodes.

In spite of the potential advantages of the wireless charging technology, little work has been reported on designing and implementing a wireless charging system for sensor networks. In this work, we design such a system that consists of (1) a mobile wireless power charger; (2) a network of sensor nodes equipped with wireless power receivers; and (3) an energy station that is responsible for monitoring the energy status of sensor nodes, deciding the power charging sequences to be executed by the mobile charger. We have built a proof-of-concept prototype of the system, and conducted experiments on the prototype to evaluate its feasibility and performance in small-scale networks. In addition, we have conducted extensive simulations to study the performance of the proposed system in large-scale networks. Experimental and simulation results demonstrate that the proposed system can utilize the wireless charging technology effectively to prolong the network lifetime. The effects of the wireless charging efficiency, the routing algorithm and various design parameters have also been studied, which may serve as useful guidelines in actual deployment of the proposed system in practice.

The rest of the paper is organized as follows. Section 2 presents the design and implementation of the proposed system. Section 3 formulates the charging problem and presents the details of charging algorithms. Sections 4 and 5 report the experimental results obtained from the prototype system, and the simulation results on large-scale networks, respectively. Finally, Section 6 summarizes the related work and Section 7 concludes the paper.

1052-8725/10 $26.00 © 2010 IEEE
DOI 10.1109/RTSS.2010.35

2 System Design and Implementation

2.1 System Overview

As shown in Fig. 1, the proposed system has three main components: a mobile charger (MC) – a mobile robot carrying a wireless power charger, a network of sensor nodes equipped with wireless power receivers, and an energy station that monitors the energy status of the network and directs the MC to charge sensor nodes.

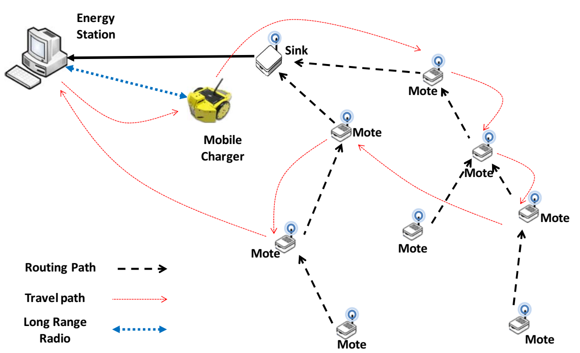


Figure 1: System overview.

The system works as follows. Sensor nodes perform application tasks such as environment monitoring, generate sensory data, and periodically report the data to the sink. In addition, they also monitor the voltage readings of their own batteries, estimate energy consumption rates, based on which derive their own lifetime, and then report the information to the sink periodically. When the energy information is forwarded to the sink, it is aggregated en-route to save communication overhead. Particularly, only the energy information of the k shortest-lifetime nodes is forwarded while the information of other nodes is dropped, where k is a system parameter. Upon receiving the energy information, the sink forwards it to the energy station, which runs a *charging algorithm* to process the information and plan the charging activities, and then sends a command message to the MC. The command includes the charging plan that the MC should execute. Once receiving the command, the MC starts charging a selected set of sensor nodes sequentially according to the instruction. When the MC receives a new command, it adjusts its charging activities accordingly.

We have built a proof-of-concept prototype of the proposed system. Its hardware and software components are described in the next two sections.

2.2 Hardware Components

The Powercast wireless power charger and receiver [12] are used in our prototype system. As shown in Fig. 2, a Powercast charger is installed on an Acroname Garcia robot [4] to become the MC, and each TelosB sensor is equipped with a Powercast receiver. When the Garcia robot moves at 1 m/s (used in our experiments), its power consumption is about 8 W and the voltage level is in the range

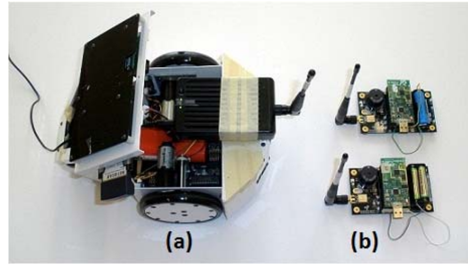


Figure 2: (a) the MC – an Acroname Garcia robot with a Powercast charger; (b) TelosB motes with Powercast receivers.

of 5.8 to 9 V. The MC communicates with the energy station (a PC in our experiments) via an IEEE 802.11b interface.

Energy charging is carried out in the 903-927 MHz band while sensor nodes communicate in the 2.4 GHz band. When the MC is charging, its power consumption is 3 W. The effective amount of power that can be captured by a receiver varies with the distance between the receiver and the MC. The relation is shown in Fig. 3, which is obtained from our field-test results. The antenna gain is 1.15 for both power charger and receiver. As shown in the figure, the receiver can receive about 45 mW power when it is 10 cm away from the charger, meaning that the charging efficiency is about 1.5%. As the distance increases, the charging efficiency decreases.

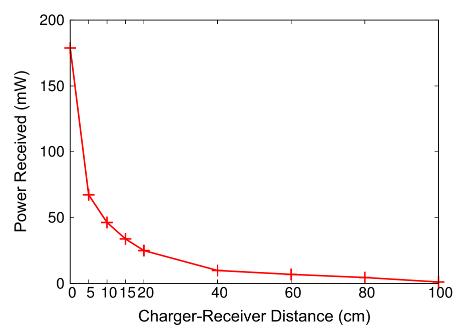


Figure 3: Power captured by the receiver vs. charger-receiver distance.

2.3 Software Components

The software components are developed based on TinyOS 2.1 [16] and loaded to each sensor node in the network. Fig. 4 shows the software architecture. The right side of the figure is the conventional framework in most sensor networks, while the left side (shaded part) is our designed Power Management component, which is compatible to and can be easily integrated with the conventional framework.

The conventional framework adopts the component-based MAC layer architecture (MLA) [7]. The energy-efficient UPMA-XMAC [15] (a variation of X-MAC [1] in TinyOS) is used as the MAC protocol. The routing layer uses the GPSR protocol. We implement a new Power Management (PM) component to monitor energy levels and consumption rates of individual sensor nodes and report such

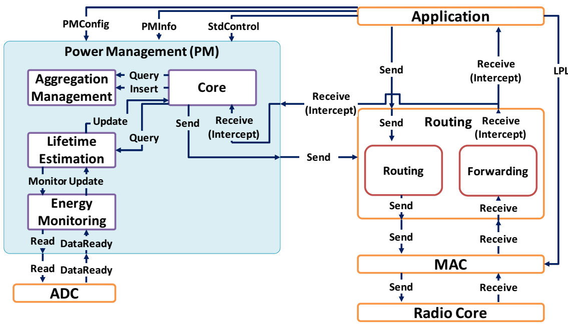


Figure 4: Software overview where the shaded part is the new Power Management component implemented in our system.

information to the energy station. The design goals of the PM component are to achieve (1) low communication and memory overheads; (2) easy integration with the conventional framework; and (3) efficient collection of accurate network-wide energy information. To integrate with the conventional framework, the PM only requires simple interfacing with the application, routing and ADC components:

- After the system is booted up, the application component needs to enable the functionality of the PM.
- The PM is allocated a collection message ID for routing the energy report message to the energy station.
- The PM intercepts forwarded energy report messages through the *intercept* interface provided by the routing layer.
- The PM uses a *read* interface to connect to the ADC component to monitor battery voltage level.

In order to adaptively tune the functionality of the PM for various application scenarios, two additional interfaces are provided to the application component: the *PMConfig* interface provides commands to configure energy information report interval, energy consumption estimation model, and low energy alarm threshold; the *PMInfo* interface provides commands to query current energy level, consumption rate and estimated lifetime.

Internally, the PM has the following modules:

- The *energy monitoring* module samples the battery voltage reading at a user-defined interval.
- The *lifetime estimation* module estimates the energy consumption rate and derives lifetime of a node based the energy information obtained from the energy monitoring module. Users are allowed to select the estimation model. The *simple moving average (SMA)* and the *exponential moving average (EMA)* models are implemented, yet new estimation models customized to specific battery characteristics (e.g., pure ultra capacitor energy source) can be added.
- The *aggregation management* module is responsible for aggregating the energy information. It records the most recent energy information about its own and its descendent nodes on the routing tree, and periodically generates an aggregated report to send to its parent node. Particularly, the aggregated report only contains the energy information about the k shortest-lifetime nodes known to this module.
- The *core* module processes intercepted report messages, triggers the aggregation management module to generate aggregated energy information reports, and updates the settings of the PM based on user's request.

3 The Charging Problem and Algorithms

As mentioned in the previous section, based on the collected energy information, the energy station runs a charging algorithm to plan the charging activities for the MC, i.e., to determine the sequence of nodes to be charged and the amount of energy to be charged to each node. In this section, we formally state the *charging problem* that the energy station tries to solve (which is NP-complete) and present two heuristic *charging algorithms* to address the issue.

3.1 Formulation of the Charging Problem

Suppose Graph $G = (V, E)$ represents the topology of a static sensor network, where each vertex stands for a sensor node and the length of an edge stands for the distance between the nodes connected by it. Suppose all sensor nodes have the same battery capacity E_s. In other words, the total amount of energy that can be stored in a sensor node's battery is E_s. For each sensor node i ($i \in V$), let e_i and cr_i denote its residual energy and energy consumption rate, respectively. There is a mobile charger (MC) in the network and its distance to each sensor node is known. The MC carries a battery of capacity E_c. When the MC charges a sensor node, it consume Λ_c power while the power received by the sensor node is $\eta\Lambda_c$ where η is called the *charging efficiency*. The MC moves at the speed of v and the power consumption for its movement is Λ_m.

We study a particular *charging problem* whose goal is to find an optimal charging sequence for the MC, denoted as $S = \langle(n_1, ct_1), \cdots, (n_{|S|}, ct_{|S|})\rangle$, where $\langle n_j, ct_j\rangle$ ($j = 1, \cdots, |S|$) represents that node n_j is charged in the j^{th} step for a period of ct_j time, such that the network lifetime is maximized. Here, the network lifetime refers to the time when the first sensor node in the network uses up its energy. Table 1 summarizes the notations used in the formulation of the charging problem.

3.2 NP-Completeness of the Charging Problem

By reducing the NP-Complete Traveling Salesman Problem (TSP) to the above charging problem, we can prove that the charging problem is also NP-Complete. The decision versions of the charging problem and the TSP problem are as follows.

Table 1: Notations Used in the Charging Problem Formulation

notation	meaning
cr_i	energy consumption rate of sensor node i
e_i	residual energy of sensor node i
E_s	battery capacity of a sensor node
E_c	battery capacity of the MC
Λ_c	the MC's charging power consumption
η	the MC's charging efficiency
Λ_m	the MC's moving power consumption
v	the MC's moving speed

- *Decision version of the charging problem*: Given a network $G = (V, E)$, the sensor node behavior (characterized by E_s, e_i and cr_i), and the MC behavior (characterized by E_c, Λ_c, Λ_m, v and η), the question is whether there exists a charging sequence S with which the network lifetime can reach at least T.

- *Decision version of the TSP problem*: Given a network $G' = (V', E')$, the question is whether there exists a visiting sequence (i.e., tour) that covers all nodes in V' and has a length of at most D.

Theorem 3.1. The decision version of the charging problem is NP-complete.

Proof. (sketch) Firstly, given a charging sequence S, we can verify whether the network lifetime can reach T with S by simulating each charging operation. This takes $O(|S|)$ steps. Therefore, the charging problem belongs to NP.

Next, we show that any instance of TSP can be reduced to an instance of the charging problem. Letting $\langle G' = (V', E'), D\rangle$ be an arbitrary instance of TSP, an instance of the charging problem $\langle G, E_s, e_i, cr_i, E_c, \Lambda_c, \Lambda_m, v, \eta\rangle$ can be constructed as follows:[1]

- $G = G'$, $cr_i = \delta$, $e_i = (D + |V|) * \delta$, $E_s = e_i + \Lambda_c$;
- $\eta = 1$, $\Lambda_m = 1$, $v = 1$, $\Lambda_c = |V| * \delta$, $E_c = \Lambda_m * D + \Lambda_c * |V|$;
- $T = D + 2 * |V|$.

Here, δ is a small positive value. The following two steps show that these two instances are equivalent.

Step I: Suppose there is a tour $\langle n_1, n_2, \cdots, n_m\rangle$ ($n_i \in V'$), which covers all node in V' and has a length of at most D, for the above TSP instance. Then, a charging sequence $S = \langle(n_1, 1), \cdots, (n_m, 1)\rangle$, where node n_i is charged in the i^{th} step for one time unit, can be constructed for the above instance of the charging problem. The charging operation is valid due to the facts that (1) the MC's energy capacity E_c is large enough to support the moving and charging operations; and (2) each node can be charged for one time unit before the capacity ceiling of its battery is reached. With this charging sequence, the lifetime of each node is extended by at least $\frac{\Lambda_c * \eta * 1}{cr_i} = |V|$. Since the sum of the MC's moving time ($\frac{D}{v} = D$) and the total charging time ($1 * |V| = |V|$) is equal to the initial nodal lifetime ($\frac{e_i}{cr_i} = D + |V|$), we know that the network can survive for at least $T = D + 2 * |V|$ time with the charging sequence. Hence, the above instance of the charging problem also has a positive answer.

Step II: Conversely, if there is a charging sequence that extends the network lifetime to $T = D + 2 * |V|$, each node must be charged for at least one time unit since the initial nodal lifetime is only $D + |V|$. This means that each node must be visited at least once. Excluding the charging energy consumption, the MC has at most $E_c - \Lambda_c * 1 * |V| = \Lambda_m * D$ energy for moving, which can be used to traverse a distance of at most D. Therefore, there exists a tour that visits every node and the length of the tour is at most D.

As the TSP is a well-known NP-complete problem, the charging problem is also NP-complete. □

3.3 Heuristic Charging Algorithms

In this section, we present two heuristic algorithms to address the difficult charging problem. As the lifetime of the network is the same as the lifetime of the first sensor node that uses up its battery energy, a *naive algorithm* is to always charge the node with the shortest lifetime to its battery capacity. Unfortunately, this may cause the MC to move back and forth between nodes, which could incur large movement overhead. The proposed *greedy* and *greedyPlus* charging algorithms aim to reduce the movement overhead.

3.3.1 Greedy Algorithm

The greedy algorithm is designed to find a charging sequence with which the lifetime of the network can be prolonged as much as possible while incurring less movement than the naive algorithm. It works as follows. All sensor nodes are sorted according to the lifetime in the ascending order. Let us denote the sorted list that contains the k shortest-lifetime nodes as $\langle(n_1, l_1), \cdots, (n_k, l_k)\rangle$, where (n_i, l_i) represents node n_i with a lifetime of l_i if it is not being charged, and $l_i \leq l_{i+1}$ ($1 \leq i \leq k-1$). Clearly, l_1 is the network lifetime if there is no charging. Then, as shown in Algorithm 1, a charging sequence can be found in the following manner:

- *Loop 1:* The algorithm tries to extend the network lifetime from l_1 towards l_2. In other words, the target network lifetime is set to l_2, and if a feasible charging sequence can be found (i.e., without violating the battery capacity of a sensor node, or depleting the battery energy of the MC in the mid of the sequence) to charge node n_1 so that the network lifetime is extended beyond l_1, the algorithm continues to *Loop 2*; otherwise, it halts. In this algorithm, the charging behavior of the MC is *greedy* in the sense that, once the MC starts charging a sensor node, it keeps charging the node for as long time as possible.

[1] In both instances, the salesman and the MC start from the same position.

- *Loop j ($2 \leq j \leq k$):* The algorithm tries to extend the network lifetime from l_1 towards l_{j+1}. Similarly, if a feasible charging sequence can be found to charge nodes n_1, n_2, $\cdots$, n_j so that the network lifetime is extended beyond l_1, the algorithm continues to the next loop; otherwise, it halts.

Algorithm 1 The Greedy Algorithm

Input:

- P: position of the MC
- α: remaining energy in the MC's battery
- $\langle (n_1, l_1), \cdots, (n_k, l_k) \rangle$: list of k shortest-lifetime nodes sorted in the ascending order of their lifetime
- $G = (V, E)$, E_s, Λ_c, Λ_m, v, η, cr_i ($i = 1, \cdots, k$): other system parameters

Output: a charging sequence

$S^* \leftarrow \emptyset$ */* the best charging sequence found so far */*
$L^* \leftarrow l_1$ */* network lifetime achieved with S^* */*
for $j = 1$ to k **do**
  if $j = k$ **then**
    $T \leftarrow \frac{E_s}{max_{1 \leq i \leq j} cr_i}$ */* target network lifetime */*
  else if $l_j = l_{j+1}$ **then**
    continue
  else
    $T \leftarrow l_{j+1}$ */* target network lifetime */*
  for each permutation of $\langle n_1, \cdots, n_j \rangle$: $\langle n'_1, \cdots, n'_j \rangle$ **do**
    $(\widetilde{P}, \widetilde{\alpha}, \widetilde{e}_{n'_1}, \cdots, \widetilde{e}_{n'_j}) \leftarrow (P, \alpha, e_{n'_1}, \cdots, e_{n'_j})$
    $S \leftarrow \emptyset, L \leftarrow 0$
    for $w = 1$ to j **do**
      ChargeTime $\leftarrow \left[(T - l_w) \times cr_{n'_w}\right] / [\eta \times \Lambda_c]$
      / time needed to charge n'_w to extend lifetime to T */*
      NodeCapTime $\leftarrow \left[E_s - \widetilde{e}_{n'_w}\right] / [\eta \times \Lambda_c]$
      / time needed to charge n'_w to full battery capacity */*
      MCCapTime $\leftarrow \left[\widetilde{\alpha} - \text{distance}(\widetilde{P}, n'_w) \times \Lambda_m / v\right] / \Lambda_c$
      / maximum time that can be spent to move to and charge n'_w with the remaining energy at the MC */*
      DeadTime $\leftarrow \min_{w+1 \leq x \leq j} \left(\frac{\widetilde{e}_{n'_x}}{cr_{n'_x}} - \frac{\text{distance}(n'_w, n'_x)}{v} \right)$
      / maximum time that can be spent to charge n'_w s.t. no other node dies before the MC can charge it */*
      $t_w \leftarrow \min(\textit{ChargeTime}, \textit{NodeCapTime}, \textit{MCCapTime}, \textit{DeadTime})$
      / t_w is the actual time that the MC will charge n'_w */*
      if $t_w < 0$ **then**
        break
      Update $\widetilde{P}, \widetilde{\alpha}, \widetilde{e}_{n'_1}, \cdots, \widetilde{e}_{n'_j}$
      (n'_w, t_w) is appended to S
    L is computed for the charging sequence S
    if $L > L^*$ **then**
      $S^* \leftarrow S, L^* \leftarrow L$
  if (no better charging sequence found in this iteration) **then**
    return S^*
return S^*

At the end of the above procedure, among all the found feasible charging sequences, the one that extends the network lifetime the most is selected by the algorithm. The complexity of this algorithm is $O(k^2 k!)$. Simulation results show that the algorithm can prolong the network lifetime effectively with a relatively small k, e.g., $k = 5$, while the performance improvement by increasing k further is not significant. Therefore, the greedy algorithm is simple to implement and effective in practice. Note that when $k = 0$, the greedy algorithm reduces to the special case when there is no charging in the network, while when $k = 1$, the greedy algorithm is equivalent to the naive algorithm.

One potential issue with the greedy algorithm is that the greedy nature of the algorithm (i.e., the MC keeps charging a sensor node for as long time as possible once started) may degrade the system performance under certain circumstances. Fig. 5 illustrates an example scenario when the greedy algorithm does not perform well. In this example, at time 0, sensor nodes n_1, n_2 and n_3 have the residual energy of 1800 J, 1800 J and 7200 J, respectively, and they have the same energy consumption rate of 0.01 J/s. The battery capacity of a sensor node and the MC is 10000 J and 270000 J, respectively. Suppose $k = 3$, $\Lambda_c = 3$ W, $\eta = 0.02$, and the MC movement cost and delay are negligible (i.e., $\Lambda_c = 0$ W, $v = \infty$).

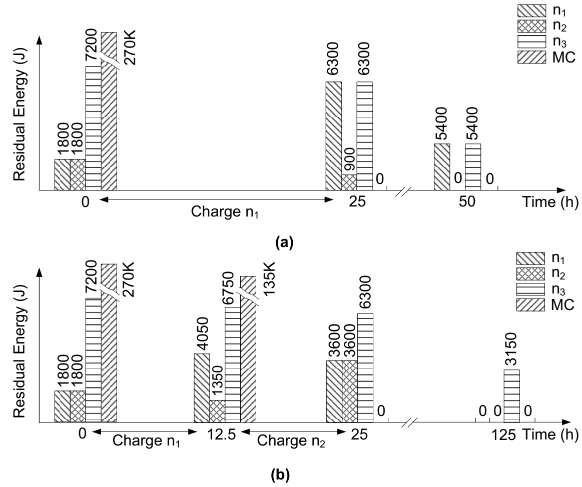


Figure 5: An example to illustrate that the greedy algorithm could be improved further: (a) network lifetime is 50 hours with the greedy algorithm; (b) network lifetime is 125 hours with more balanced charging.

As shown in Fig. 5(a), with the greedy algorithm, the MC starts charging n_1 at time 0 and continues to time 25 hours when the residual energy of n_1 becomes the same as that of n_3 (hence they have the same lifetime as they have the same energy consumption rate). This action causes the MC to use up all of its battery energy. As a result, the network lifetime is 50 hours when the residual energy of n_2 becomes zero. In comparison, if the MC takes a less greedy approach that charges n_1 for 12.5 hours and then charges n_2 for 12.5 hours, as shown in Fig. 5(b), the network can survive much longer for 125 hours in total, resulting in a 250%

performance improvement. Motivated by this observation, we revise the greedy algorithm to allow the MC to charge sensor nodes in a more balanced manner. The revised algorithm is named the *greedyPlus algorithm*.

3.3.2 GreedyPlus Algorithm

In the greedy algorithm, the MC tends to charge a node in a greedy manner towards the target lifetime T without considering whether the lifetime of other nodes can also be extended to T. In fact, as long as one node cannot have its lifetime extended to T, the network lifetime cannot reach T, meaning that some of the energy being charged to the current node may be wasted. The greedyPlus algorithm improves upon the greedy algorithm by applying binary search to find a more suitable target network lifetime at each loop, which is a target achievable by all sensor nodes in the network. Fig. 6 shows the flowchart of the greedyPlus algorithm, which is based on the pseudo-code of the greedy algorithm in Algorithm 1. In the flowchart, ϵ is a small quantity to help define exit conditions for the greedyPlus algorithm.

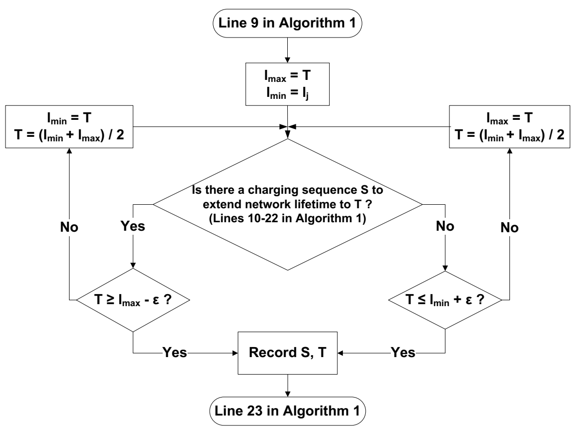


Figure 6: GreedyPlus algorithm Flowchart.

4 Experiments

Experiments have been performed on the prototype system to evaluate its feasibility and performance.

4.1 Experimental Setup

Nine Telosb sensor nodes are used in the experiments. Each node is powered by two 1.5V 2000mAh Alkaline rechargeable batteries. The power receiver in the node can charge the batteries when it receives energy from the MC, and the charger-receiver distance varies from 5cm to 20cm.

The sensor nodes are deployed in the line or grid topology as shown in Fig. 7, where neighboring nodes are two meters apart and the CC2420 radio transmission power is set to level 3 which results in a communication range of about 3.5 meters. In both topologies, node 1 works as the sink connected to a PC with stable power supply and therefore it does not need to be charged.

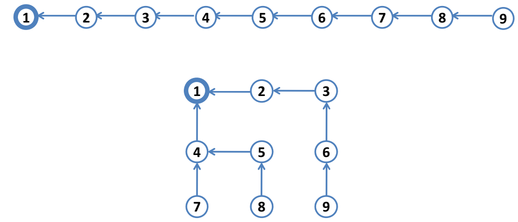


Figure 7: Experimental topologies.

Before each experiment, the batteries on each node are pre-charged to a certain voltage level up to 2.9 V (normalized to energy level 100% in the following figures), and the energy of a mote is assumed to be depleted when the voltage level of its batteries drops to 2.7 V (normalized to energy level 0% in the following figures).

During the experiments, each sensor node sends out a data packet every 16 $\sim$ 20 seconds and an energy report every 10 minutes. The UPMA-XMAC protocol running on each sensor node sets its low power listening interval to 2 seconds. Every 10 minutes, the energy station runs the charging algorithm to adjust the charging plan.

4.2 Evaluation Results

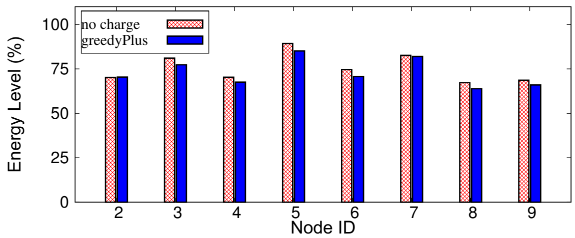
Fig. 8 and Fig. 9 show the evaluation results for the line and grid topologies, respectively. We measure (1) the initial energy level of individual nodes; (2) the lifetime of the network and individual nodes when there is no energy charging (tagged as *no charge* in the figures) and when the greedyPlus algorithm is used; and (3) the distribution of charging time among individual nodes.

4.2.1 Evaluation Results for Line Topology

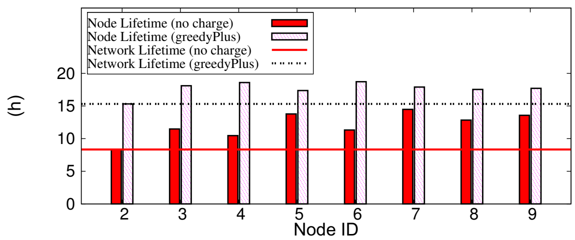
As shown in Fig. 8(b), the greedyPlus algorithm significantly improves the network lifetime from 8.3 hours to 15.3 hours, an increase of 84%. This is accomplished by charging more energy to nodes that have shorter lifetime if there were no charging. Particularly, nodes 2, 3 and 4 have higher energy consumption rates than other nodes because they forward more data packets; node 6 has low energy level at the beginning of the experiment. Hence, these nodes have shorter lifetime than others when there is no charging. Their lifetime (especially the lifetime of node 2) becomes the bottleneck of the network lifetime. As shown in Fig. 8(c), the greedyPlus algorithm charges more energy to these nodes to extend their lifetime and consequently improve the network lifetime.

4.2.2 Evaluation Results for Grid Topology

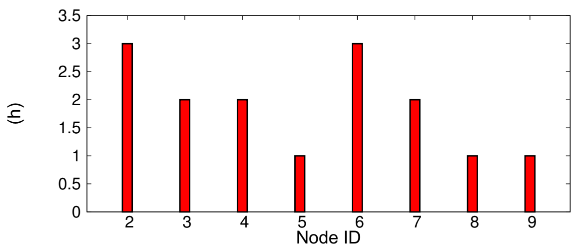
For the grid topology, as shown in Fig. 9(b), the greedyPlus algorithm improves the network lifetime from 7.2 hours to 15.92 hours, an increase of 120%. Nodes 2, 4 and 6 are charged the most as they have a higher energy consumption rate or a lower initial energy level. Node 5 is seldom charged due to its high initial energy level and node 7 is



(a) Initial energy of individual nodes.

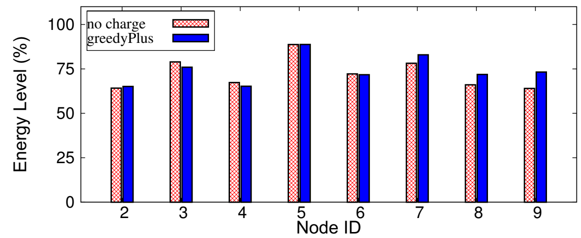


(b) Lifetime of the network and individual nodes.

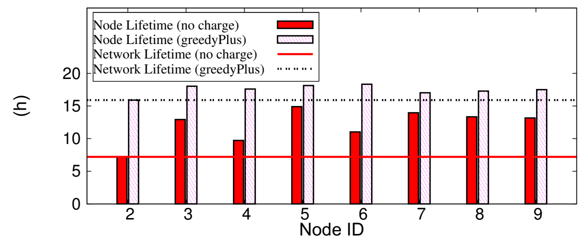


(c) Distribution of charging time among sensor nodes.

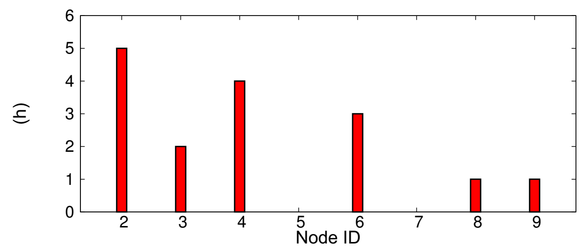
Figure 8: Evaluation results for line topology.



(a) Initial energy of individual nodes.



(b) Lifetime of the network and individual nodes.



(c) Distribution of charging time among sensor nodes.

Figure 9: Evaluation results for grid topology.

seldom charged for its being a leaf node with low energy consumption rate. Also being leaf nodes, nodes 8 and 9 however receive more charging than node 7 because they have less initial energy than node 7.

4.2.3 Summary

The evaluation results have demonstrated that, in a sensor network where motes run on different initial energy supplies and heterogeneous consumption rates, the proposed wireless charging system can improve the network lifetime through accurately identifying the bottleneck nodes and charging them to extend their lifetime.

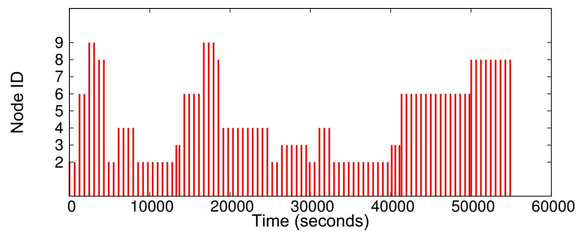
4.3 Discussion

The experiments not only verify the feasibility and effectiveness of our proposed design, but also demonstrate some phenomena that are hard to discover through theoretical derivations or simulations. Such discovery is helpful in enhancing our design and making it more effective in practice. As an example, we present in the following the phenomenon of abnormal voltage level reading which happens during and immediately after a node being charged. We also analyze its effects on the charging performance, and describe our solution to address the issue.

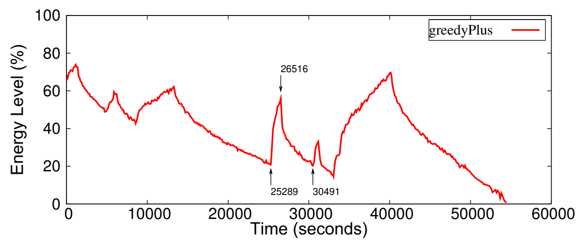
Fig. 10(a) displays a charging trace in one experiment. Corresponding to this charging trace, Fig. 10(b) demonstrates the changes of node 2's voltage readings. Fig. 10(c) shows the changes of node 2's voltage readings when there is no charging. It can be observed in Fig. 10(b) that, when node 2 is being charged (e.g., during time $25289 \sim 26516$ seconds), its voltage reading increases very fast. However, the fast increase is "false" and not stable because, as also shown in Fig. 10(b), immediately after the charging phase ends (e.g., during time $26516 \sim 30491$ seconds), the voltage reading drops much more rapidly than the normal case shown in Fig. 10(c).

If such instant measurements of energy levels are used directly as the input to the charging algorithm, the algorithm may output less efficient charging decisions. Specifically, a measured fast but false increase of energy level may mislead the algorithm to believe that the currently charged node has been charged with sufficient amount of energy and decide to charge another node; but immediately after the MC leaves, the fast drop of energy level will be measured, which may force the charging algorithm to schedule the MC back to charge the node again. The back-and-forth scheduling of the MC can waste energy and hence degrade the performance of the charging system.

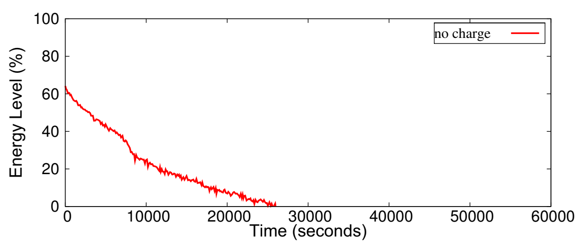
To address this issue, in our design and implementation, when a node is being charged, its energy level that is just measured is not used directly as the input to the charging



(a) Charging trace.



(b) Node 2's energy variation with charging trace in Fig. 10(a): its voltage reading increases fast when being charged and drops fast immediately after charging completes.



(c) Node 2's energy variation without Charging.

Figure 10: Abnormal voltage reading during and immediately after a node being charged.

algorithm. Instead, its energy levels measured during a certain recent time frame (e.g., the past 10 minutes) are averaged and then scaled down by another certain constant (e.g., 100) to remove the abnormally sharp increase of its voltage reading during the charging time. According to the experiments, the benefit brought by this averaging and scaling technique is significant. Particularly, the average improvement of network lifetime is about 84% and 120% for the line and grid topologies respectively when the technique is adopted, while the improvement is only about 40% if the technique is not used.

5 Simulations

Extensive simulations have been conducted to evaluate the proposed system in large-scale networks.

5.1 Simulation Setup

The proposed system is simulated in a custom simulator. In the simulations, 100 nodes are randomly deployed in a 500 m $\times$ 500 m field. Every sensor node is a data source which reports data packets to the sink at the rate of λ. It also sends one energy report every hour; hence, the energy station runs the charging algorithm once every hour.

The routing algorithm adopted in the simulations uses metric $C_i = Tr_i * u^{1-\frac{e_i}{E_s}}$ [9] to select routes. In the metric, u is a system parameter, e_i is node i's residual energy and Tr_i is the sum of energy consumption for packet transmission and reception at node i. This metric is a combination of the minimum energy (Tr_{ij}) and max-min residual energy ($u^{1-\frac{e_i}{E_s}}$) metrics. When $u = 1$, the metric is reduced to the minimum energy (*ME*) metric which is used to find the path that can minimize the network-wide energy consumption; when $u > 1$, it is reduced to an energy-aware (*EA*) metric that aims to balance energy consumption among all nodes in the network. We let $u = 1$ or $u = 100$ in the simulations. Table 2 lists other default simulation parameters.

Table 2: Default Simulation Parameters

parameter	value
communication range of a sensor node(m)	70
battery capacity of a sensor node: E_s (KJ)	10
battery capacity of MC: E_c (KJ)	2000
data packet generation rate: λ (packets/hour)	12
the MC's charging power consumption: Λ_c (W)	3
the MC's moving power consumption: Λ_m (W)	50
the MC's moving speed (m/s)	1
the sensor's tx power consumption (J/packet)	0.05
the sensor's rx power consumption (J/packet)	0.06
system parameter k	5
the MC's charging efficiency: η (%)	1.5
the number of sinks	4
the routing metric	ME

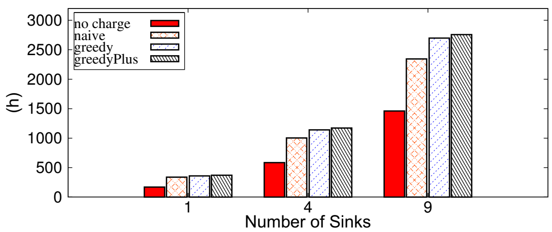
5.2 Simulation Results

We measure the network lifetime of the naive, greedy and greedyPlus algorithms under different situations by varying the number of sinks, the system parameter k and the charging efficiency η.

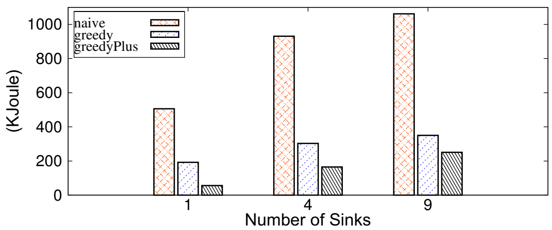
5.2.1 Network lifetime with varying number of sinks

We first measure the lifetime when the number of sinks changes. If there are n sinks, the whole network field is divided evenly into n areas and one sink is placed at the center of each area. Fig. 11(a) shows that all three algorithms can significantly improve the network lifetime (by at least 80%) regardless of the number of sinks. Among these algorithms, the greedy and greedyPlus algorithms outperform the naive one as they can significantly reduce energy consumption on MC's movement, which is demonstrated in Fig. 11(b), and therefore can use more energy to charge sensor nodes.

It is also found that, as the number of sinks increases, less improvement of network lifetime is achieved by the charging algorithms. Particularly, the greedyPlus algorithm extends the network lifetime by 117% when there is one sink,

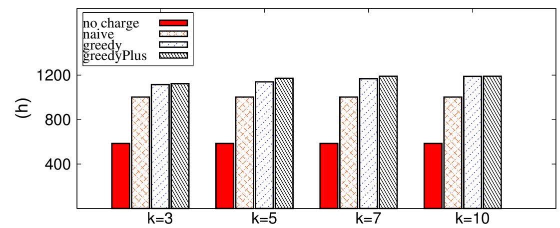


(a) Network lifetime.

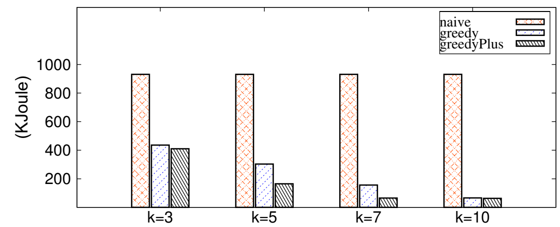


(b) Energy consumed by MC movement.

Figure 11: Effects of the number of sinks.



(a) Network lifetime.



(b) Energy consumed by MC movement.

Figure 12: Effects of system parameter k.

but the ratio drops to 88% when there are 9 sinks. This is due to the following reasons. The energy consumption rates of different sensor nodes become more even as the number of sinks increases. With the increase, more sensor nodes need to be charged to extend the network lifetime. As the MC's charging capacity and efficiency are bounded, increasing the number of charged sensor nodes decreases the amount of energy charged to each of these nodes. Consequently, the overall improvement in network lifetime decreases. The improvement of network lifetime is further reduced because the MC has to consume more energy in movement as it needs to charge more nodes, which decreases the amount of energy that can be used for charging sensor nodes. The phenomenon indicates that, the less even are the energy consumption rates among sensor nodes, the fewer sensor nodes need to be charged and the better performance can be achieved by the charging algorithms.

5.2.2 Network lifetime with varying k

The network lifetime extended by the greedy and greedyPlus algorithms does not increase linearly or significantly as k increases. This phenomenon is attributed to the following reasons. If the number of bottleneck nodes that constrain the network lifetime is less than k, increasing k does not improve the network lifetime as only the energy information about bottleneck nodes is useful for charging planning. If the number of bottleneck nodes is larger than k, the information of all the bottleneck nodes can still be gradually obtained and considered by the charging algorithms as the algorithms are run once every certain time interval and each running of the algorithm is based on the energy information of the k shortest-lifetime sensor nodes at the moment. As shown in Fig. 12(b), a larger k does help in reducing the movement energy consumption as the information of more nodes is considered in charging planning. However, the benefit brought by increasing k is not significant and the benefit decreases as k gets large. In the simulations, we let $k = 5$ by default.

5.2.3 Network lifetime with varying η

Due to the relatively low value of η (i.e., 1.5%) in previous simulations, even when all of $E_c = 2000$ KJ energy is used by the MC for charging, only 30 KJ can be received by sensor nodes. Hence, the charging efficiency has been a major constraint on improving the network lifetime. Fig. 13 shows that the network lifetime can be significantly increased as η increases. Particularly, the greedyPlus algorithm can extend the network lifetime by about 100% with $\eta = 1.5\%$ and the extension ratio rises to 200% when $\eta = 6\%$.

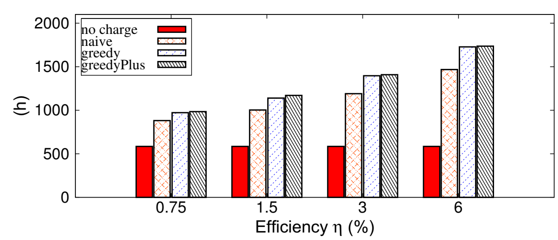
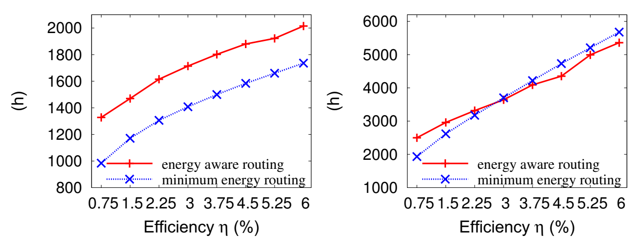


Figure 13: Effects of charging efficiency η.

The charging efficiency can be increased through advance in charging technology. In fact, it can be increased through delicate sensor node deployment such as the aggregated sensor node deployment strategy proposed by Tong *et al.* [18]. Thus, when combined with the aggregated node deployment strategy, the performance of the proposed wireless charging system may be improved further.

5.2.4 Energy-efficient Routing vs. Energy-balanced Routing

When using the ME metric, the routing algorithm adopted in the simulations becomes an energy-efficient algorithm as it tends to find routes that consume the least total energy. When using the EA metric, the routing algorithm becomes an energy-balanced algorithm as it tends to find routes that distribute communication workload among all sensor nodes as evenly as possible. In a non-rechargeable network, the energy-balanced routing algorithm outperforms the energy-efficient algorithm in terms of network lifetime, but this may not be always true when sensor nodes can be charged. Extensive simulations have been conducted to study which of these two types of algorithms is more beneficial to our proposed system.



(a) $E_c = 2000$ KJ. (b) $E_c = 20000$ KJ.

Figure 14: Effects of the routing algorithm.

In our simulations, E_c (the total amount of energy carried by the MC) and η vary, the greedyPlus algorithm is run, and the achieved network lifetime is measured. The results are shown in Fig. 14. As shown in the figure, the energy-balanced algorithm (EA) outperforms the energy-efficient algorithm (ME) when E_c or η is small; however, ME outperforms EA when both E_c and η are large. The reasons behind the phenomena are as follows.

When the ME routing algorithm is used, nodes that are on multiple routing paths have higher communication overhead and thus become bottleneck nodes. When η or E_c is small, some of these bottleneck nodes cannot be recharged in a timely manner before their energy is used up though some other nodes may still have lots of energy left, which results in a short network lifetime. On the other hand, the EA routing algorithm tends to balance the energy consumption among nodes. With EA, the energy consumption rates of bottleneck nodes typically are lower than the bottleneck nodes in the network running ME, and hence their lifetime is also longer. Therefore, the network running EA can achieve a longer lifetime.

When E_c and η are large, bottleneck nodes in the network running ME are likely to be charged promptly. Hence, the charging algorithm can effectively balance the lifetime between bottleneck and non-bottleneck nodes as the EA routing algorithm does. Moreover, the ME algorithm consumes less network-wide energy than the EA algorithm. Resulted from these two effects, the network running ME can achieve longer lifetime than the one running EA.

5.2.5 Network lifetime with varying λ

Fig. 15 demonstrates how much the network lifetime can be improved by the greedyPlus algorithm, as the date rate (λ) varies. When λ is small (e.g., $\lambda < 48$ packets per hour), the lifetime improvement is over 100% for both $E_c = 20000KJ$ and $E_c = 2000KJ$. However, the improvement ratio decreases as λ increases. This is due to the fact that the charging capability of the MC is determined by $\Lambda_c * \eta$ (the maximal amount of energy that can be actually charged to the network per unit of time) and E_c (the total amount of energy carried by the MC). When λ is small, $\Lambda_c * \eta$ is large enough to keep the network alive before the MC runs out of its energy at time $\frac{E_c}{\Lambda_c}$. Hence, E_c becomes the major factor that determines the improvement of network lifetime, and a larger E_c brings more improvement in network lifetime. On the other hand, when λ is large, $\Lambda_c * \eta$ becomes the major factor that determines the improvement of network lifetime, as some bottleneck nodes may die much earlier before the MC runs out of its energy. In this case, increasing E_c does not help in prolonging the network lifetime.

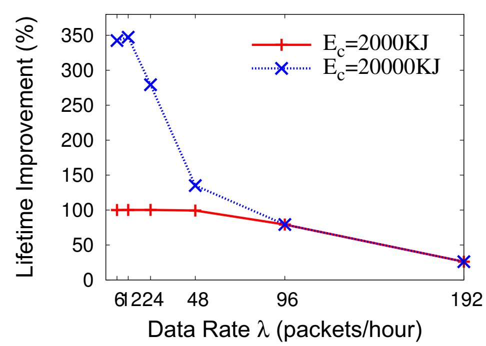


Figure 15: Effects of λ on the performance of the greedyPlus algorithm.

5.2.6 Summary

To summarize, we have the following observations from simulation results:

- Wireless charging is effective in prolonging the network lifetime.
- Through careful movement planning, the proposed greedy and greedyPlus algorithms consume less energy for movement and thus have more energy for charging to extend the network lifetime. Though these algorithms have a complexity exponential with system parameter k, a small k can generate decent results. Hence, these algorithms are practically efficient.
- The improvement of the network lifetime becomes more significant as the charging efficiency increases. This observation is encouraging as the charging efficiency can be improved through not only advance

in charging technology but also combining our proposed system with the aggregated node deployment strategy [18].

- With the proposed system, the energy-balanced routing algorithm may not always outperform the energy-efficient algorithm in terms of prolonging the network lifetime. In general, when the MC is powerful (i.e., large E_c and η), the energy-efficient routing algorithm is more beneficial; when the MC has small E_c or η, the energy-balanced routing algorithm is more beneficial.

- The network lifetime improved by the proposed system is bounded by two factors: $\Lambda_c * \eta$ and E_c. When the data generation rate at sensor nodes is low, the energy carried by the MC (E_c) determines the network lifetime improvement, while for sensor network applications with high data rates, the charging power (Λ_c) and efficiency (η) of the MC become the more dominant factor in determining the network lifetime improvement.

6 Related Work

Recent research work on harvesting environmental energy such as solar [3, 6], wind [11], thermal [14] and vibration [5] have shown great promise of addressing the battery energy limitation problem in the sensor network. The work presented in this paper is different from these previous results in that we employ a controllable energy source (i.e., the mobile charger) to charge sensor nodes. Therefore, we can proactively adjust the charging pattern rather than passively adapt to the environmental energy availability. Tong *et al.* [17, 19] proposed a node reclamation and replacement strategy (NRR) which makes use of a mobile repairman to periodically replace low or no energy sensors with full energy ones. The reclaimed sensors are brought back to the base station for recharging. The NRR scheme requires the robot to be able to locate the sensor position accurately enough for replacement. On the other hand, the wireless charging technique used in this paper is more tolerant to localization inaccuracy since it can charge a node from a distance range.

Another important research area concerns exploiting sink mobility to alleviative the traffic aggregation burden from a fixed set of sensor nodes near the sink to the peripheral nodes [10, 13]. Our work tries to extend the system lifetime from a different perspective by taking advantage of a mobile charger (sink is static) rather than a mobile sink.

7 Conclusion

We propose a wireless charging system for sensor networks. We present the design and implementation of the system, especially study the charging planning problem and propose several heuristic algorithms. Experiments are conducted on an implemented prototype of the proposed system to evaluate the feasibility and the effectiveness of the system on prolonging the sensor network lifetime. Simulations are also conducted to evaluate the performance of the proposed system in large-scale networks and to study the effects of various configuration and design parameters. The results verify that the proposed system can extend the sensor network lifetime significantly.

References

[1] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *SenSys '06*, pages 307–320, 2006.

[2] J. H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM '05*, pages 22–31, 2000.

[3] K.-W. Fan, Z. Zheng, and P. Sinha. Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks. In *SenSys '08*, pages 239–252, 2008.

[4] Garcia. http://www.acroname.com.

[5] S. M. Jose, J. O. Mur-mir, R. Amirtharajah, A. P. Ch, and J. H. Lang. Vibration-to-electric energy conversion. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(1):64–76, Feb 2001.

[6] A. Kansal, D. Potter, and M. B. Srivastava. Performance aware tasking for environmentally powered sensor networks. *SIGMETRICS Perform. Eval. Rev.*, 32(1):223–234, 2004.

[7] K. Klues, G. Hackmann, O. Chipara, and C. Lu. A component based architecture for power-efficient media access control in wireless sensor networks. In *SenSys '07*, November 2007.

[8] Q. Li, J. Aslam, and D. Rus. Online power-aware routing in wireless ad-hoc networks. In *MobiCom '01*, pages 97–107, 2001.

[9] L. Lin, N. B. Shroff, and R. Srikant. Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources. *IEEE/ACM Transaction on Networking*, 15(5):1021–1034, 2007.

[10] J. Luo and J. P. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *INFOCOM '05*, March 2005.

[11] C. Park and P. Chou. Ambimax: Autonomous energy harvesting platform for multi-supply wireless sensor nodes. In *SECON '06*, pages 168–177, Sept. 2006.

[12] Powercast. http://www.powercastco.com.

[13] Y. Shi and Y. T. Hou. Theoretical results on base station movement problem for sensor network. In *INFOCOM '08*, April 2008.

[14] I. Stark. Invited talk: Thermal energy harvesting with thermo life. In *BSN '06*, pages 19–22, April 2006.

[15] Y. Sun, O. Gurewitz, and D. B. Johnson. RI-MAC: A receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *SenSys '08*, 2008.

[16] TinyOS. http://www.tinyos.net/.

[17] B. Tong, Z. Li, G. Wang, and W. Zhang. On-demand node reclamation and replacement for guaranteed area coverage in long-lived sensor networks. In *QShine '09*, November 2009.

[18] B. Tong, Z. Li, G. Wang, and W. Zhang. How wireless power charging technology affects sensor network deployment and routing. In *ICDCS '10*, June 2010.

[19] B. Tong, G. Wang, W. Zhang, and C. Wang. Node reclamation and replacement for long-lived sensor networks. In *SECON '09*, June 2009.