2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems

NETWRAP: An NDN Based Real Time Wireless Recharging Framework for Wireless Sensor Networks

Ji Li[1], Cong Wang[1], Fan Ye[2], and Yuanyuan Yang [1]
[1]*Dept. of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA*
[2]*School of Electronics Engineering and Computer Science, Peking University, P.R. China*

***Abstract*—A mobile vehicle equipped with wireless energy transmission technology can move around a wireless sensor network and recharge nodes over the air, leading to potentially *perpetual operation* if nodes can always be recharged before energy depletion. When to recharge which nodes, and in what order, critically impact the outcome. So far only a few works have studied this problem and relatively static recharging policies were proposed. However, dynamic changes such as unpredictable energy consumption variations in nodes, and practical issues like scalable and efficient gathering of energy information, are not yet addressed. In this paper, we propose NETWRAP, an NDN based Real Time Wireless Recharging Protocol for dynamic recharging in wireless sensor networks. We leverage concepts and mechanisms from NDN (*Named Data Networking*) to design a set of protocols that continuously gather and deliver energy information to the mobile vehicle, including unpredictable emergencies, in a scalable and efficient manner. We derive analytic results on *energy neutral* conditions that give rise to perpetual operation. We also discover that optimal recharging of multiple emergencies is an Orienteering problem with Knapsack approximation. Our extensive simulations demonstrate the effectiveness and efficiency of the proposed framework and validate the theoretical analysis.**

***Index Terms*—Wireless sensor networks, named data networking, wireless recharging, energy efficient designs, perpetual operation.**

I. Introduction

Wireless energy transmission techniques [1], [2] have great potential to prolong the lifetime of wireless sensor networks. With such techniques, the energy of wireless sensor nodes can be replenished over the air without any wire or plug, and more reliable energy sources can be provided than those from environmental energy harvesting techniques [4], [5]. A mobile energy replenishing vehicle (called *SenCar*) can move around and recharge nodes conveniently. The recharging policy - when to recharge which nodes and in what order - critically impacts the efficiency and thus the lifetime of the network.

So far only a few work [8], [9] has studied the recharging policy problem. Basically, nodes report their energy levels periodically, and a centralized algorithm computes a specific order to recharge all nodes in the next cycle. Though commendable first steps, they do not fully consider important practical issues, which severely limit their applicability in a real environment.

First, it takes nontrivial (e.g., 30-60 min) time to recharge a commercial off-the-shelf battery, such that finishing one round of recharging for a network of a few hundred nodes may take several days. During this time the energy levels of nodes may have changed significantly due to unpredictable external events that can trigger extensive activities and quickly drain the battery. The recharging policy computed at the beginning of the cycle is no longer optimal. This can cause energy depletion on some nodes, leading to network disconnection or application failures. Second, the timely, efficient and scalable gathering of energy information of nodes to a mobile vehicle is an important and challenging issue in itself. The previous work does not consider it and assumes such information is already available. Finally, they use centralized algorithms that have high complexity and may not scale to large networks. A distributed solution is more desirable in real network environments.

In this paper, we propose a novel real time recharging framework that optimizes the recharging policy under dynamic network conditions. Instead of nodes reporting their energy levels only after a long period, a scalable and efficient energy information aggregation protocol gathers battery levels continuously from all sensor nodes upon requests by the SenCar. The SenCar receives such information and makes recharging decisions based on the latest energy information. To deal with unpredictable emergencies where nodes may dramatically drain the battery in short time, the recharging of sensor nodes whose energy levels are below a critical threshold has high priority and takes precedence over those that can work for relatively long time with their residual energy.

We apply *Named Data Networking (NDN)* [10] techniques to gather and deliver energy information to the SenCar. To scale to large networks, we divide the network in a hierarchical fashion and the energy information is aggregated bottom-up through different levels. NDN uses names instead of locations to address data, which is a natural match for aggregated energy information that belongs to an area instead of any particular node. Thus the aggregated energy information can be addressed by the area's name. NDN also supports mobile receivers because the routing states in intermediate nodes are constantly updated to follow the movements of receivers. This is important for the SenCar to receive the energy information timely after it changes its location.

We also study the conditions for perpetual operations of the network under such a recharging framework. We identify the energy neutral requirement (i.e., the energy replenished should be more than or equal to the energy consumed) and formally derive the probability for this condition to hold in a network. Given the network and SenCar parameters, an administrator can estimate the likelihood that the network can operate perpetually.

Due to the unpredictable nature of external events, the SenCar may need to deal with multiple concurrent *emergencies* occurring in different locations. We study their optimal recharging policy and find that it can be formulated as an *Orienteering problem* [13], which has been studied before but only with computationally intensive solutions. We show that by taking reasonable approximations, it can be converted into a Knapsack problem, which has more efficient heuristics.

978-0-7695-5104-3/13 $26.00 © 2013 IEEE
DOI 10.1109/MASS.2013.29

We make the following contributions in this paper. First, we propose a novel real time recharging framework for wireless sensor networks, consisting of a set of scalable and efficient NDN based energy aggregation and gathering protocols. The protocols satisfy both normal and emergency recharging needs for a mobile vehicle. To the best of our knowledge, this is the first work capable of adapting to dynamic network conditions such as emergencies, and the first effort to apply NDN techniques to wireless sensor networks. Second, we formally define the energy neutral condition and analyze the theoretical probability for perpetual operations; we also discover that optimal recharging of multiple emergencies is an Orienteering problem, and further approximate it to a Knapsack problem with more efficient heuristics. Finally, we compare our real-time framework with the static approaches in [8], [9] and conduct extensive simulations to demonstrate the effectiveness and efficiency of the proposed framework. We validate the energy neutral analysis results, and show that the Knapsack approximation is within 99% of the exact solution under typical network parameters.

The rest of the paper is organized as follows. Section II discusses the related work. Section III outlines the framework and assumptions made in the network model. Section IV describes the operations and mechanisms of our protocol followed by Section V on deriving the energy neutral conditions and solving the optimal emergency recharging problem. Finally, Section VI shows simulation results and Section VII concludes the paper.

II. Related Work

A. Wireless Rechargeable Sensor Networks

Wireless rechargeable sensor networks have drawn interest from both academia and industry recently [6]–[9]. In [6], the impact of wireless charging technology on wireless sensor networks was investigated, and heuristic algorithms were developed to solve the deployment and routing problem. In [7], the problems of point provisioning and path provisioning were studied in a wireless rechargeable sensor network built from industrial wireless sensing platform and commercial off-the-shelf RFID readers. In [9], the scenario of periodically recharging each sensor node using a mobile charging vehicle was considered. A near-optimal solution was provided to calculate the optimal traveling path of the mobile car. It forms the shortest Hamilton cycle through all sensor nodes and maximizes the ratio of mobile car's charging to idle time. In [8], the problem of joint optimization of effective energy charging and high-performance data collections with bounded tour length and data gathering interval was studied. A two-step approach was proposed to recharge the nodes with the least residual energy and maximize network utility.

The above work makes pioneering steps in this new area of wireless rechargeable sensor networks. However, none of them consider how the energy information is aggregated and reported to the mobile car, nor do they handle changes in energy levels that occur inevitably and unpredictably during long recharging cycles.

B. Named Data Networking (NDN)

Named Data Networking is a new network architecture proposed recently for the Internet [10]. In NDN, data are addressed by their names instead of hosting nodes' locations. The operation is based on two types of messages, *Interest* and *Data*, and the communication is initiated by the receiver. A receiver interested in certain data sends Interest messages carrying the name of the desired data. The Interest message propagates in the network following FIB (Forward Interest Base) states towards nodes hosting desired data. It also leaves a "trail" of PIT (Pending Interest Table) states in intermediate nodes. Once the Interest reaches a node hosting the desired data, Data messages can follow PIT states to traverse back to the receiver. So far NDN research has largely focused on the Internet, with some efforts on mobile networking. Whether it can be used to satisfy the needs of wireless sensor networks is still unexplored. In this paper, we use wireless recharging as a case study to investigate its applicability in wireless sensor networks, and we find that it does have attractive benefits in our scenario.

III. A Novel Framework for Wireless Rechargeable Sensor Networks

In this section, we describe the components, network model and assumptions for our NDN based wireless recharging framework. NDN has a few attractive benefits for our environment. First, by sending out new Interest packets, a mobile receiver can continuously update the routing states (i.e., PIT entries) in intermediate nodes. Data can follow the reverse paths traversed by the most recent Interest packets and reach the new location of the receiver. This solves the mobility issue of SenCar and ensures that the latest energy information can reach SenCar in a timely manner. Second, to scale to large networks, we divide the network in a hierarchical fashion and energy information is gathered in aggregated forms. Thus data is bounded to an area rather than any particular node. This makes a natural fit for NDN: data can be addressed using the area's name.

A. Network Components

The network consists of the following components for building and maintaining the name hierarchy, querying energy information, and recharging the sensor nodes.

SenCar: The SenCar queries the network for energy information and recharge nodes based on the energy information collected.

Head node: A head is a sensor node delegated to aggregate energy information from its subordinate area. When requested by the SenCar (for top level heads) or by the head of the upper level (for other heads), a head queries energy information from subordinate sub-areas at the lower level, aggregates such information and sends to the requester.

Proxy: A proxy node aggregates emergencies from sensor nodes and reports such information to the SenCar when queried. Only top level head nodes are proxies.

Normal Node: A sensor node not selected as a head is a normal node. It reports energy information to head nodes, or

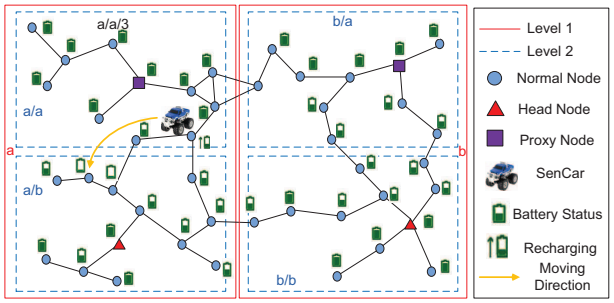

Fig. 1. An illustration of area names and network components.

sends emergency directly to its proxy when its energy level drops below the emergency recharge threshold.

B. Name Assignments and Network Model

We assume sensors are scattered uniformly randomly. The network field is divided into several areas and each area is further divided recursively. The division of the area is based on geographical coordinates of the sensing field. Each division generates some new *sub-areas* and increases the number of *levels* in the network. This process repeats until the bottom level subarea becomes small enough such that the SenCar can recharge one such a subarea with no need to query normal energy levels in the middle of the recharging process. Fig. 1 gives an example of a 2-level network with 2 areas (in red solid line), each further split into 2 sub-areas (in blue dash line). Each sub-area on the second level contains about 10 sensor nodes.

Based on the results of area divisions, we assign NDN data names for different subareas in a hierarchical manner. For example, Fig. 1 shows all the name assignments for different subareas (e.g., the first level areas are "a" and "b", and the second level has "a/a", "a/b", ...). Thus each subarea is identified by its unique hierarchical name. Each node has an ID including the name of the containing bottom level subarea plus an identifier. For example, "a/a/3" is node "3" in subarea "a/a". A message can carry the name of intended propagation subarea and nodes can use their names to ensure that the propagation does not go beyond that subarea.

In addition, we have the following assumptions: 1) Sensor nodes are stationary and each node knows its location. 2) Nodes have the same transmission range and messages are forwarded over multiple hops in large networks. 3) The SenCar has a positioning system and knows its own location. The IDs and locations of all sensor nodes, and the subarea names are known to the SenCar (e.g., through an one-time effort at the initial stage). 4) The field is barrier-free so the SenCar can move to any sensor node in straight movement lines. 5) Sensors might perform different tasks so the energy consumption is not uniform among nodes.

IV. The NDN based Real Time Wireless Recharging Protocol

In this section, we present the detailed design of NDN based Real Time Wireless Recharging Protocol (NETWRAP). We first give an overview of NETWRAP in Section IV-A. Then we describe different operating phases of NETWRAP in Section IV-B.

A. Protocol Overview

In NETWRAP, the SenCar obtains the most up-to-date energy information from sensors and makes recharge decisions in real time. The energy information is aggregated on heads at different levels. To be robust, the head is usually selected as the node having the maximum energy level in its area. This selection process is done at the beginning of network startup through the propagation of *head selection* messages. The details will be discussed in the next subsection.

To start a new round of energy information collection, the SenCar sends out an *energy interest* message to poll the *heads* on the top level. Once the heads receive such messages, they send lower level *energy interest* messages to their child heads in respective subordinate areas. This process repeats down in the head node hierarchy, until finally the bottom level *energy interest* messages reach the nodes in the bottom level subareas.

Once a sensor node receives a bottom level *energy interest*, it responds by sending out an *energy* message containing its ID and battery level. When the heads on the bottom level receive such *energy* messages, they select sensor nodes below a normal recharge threshold, and send the names of these nodes and their energy information in an *energy* message to their parent head nodes. This is repeated up the head node hierarchy, until finally the top level head nodes have the aggregated energy information and send it to the SenCar.

Note that in NETWRAP, for the purpose of reducing transmission overhead, the head is delegated partial responsibilities to pre-select sensor nodes to be recharged. This is done as the heads select the nodes with low energy level. In upper levels, a head selects the subordinate area which can be recharged with the most amount of energy. Thus the SenCar can replenish the network with more energy in one movement.

Such normal energy aggregation is conducted at the request of the SenCar. For emergency nodes that have dangerously low battery levels below an emergency threshold, they send out *emergency* messages to the proxy that manages its area. The route to its proxy is built by *head selection* messages from the proxy.

The SenCar monitors whether there is any emergency by sending out *emergency interest* messages after finishing recharging any single node. These messages are directed to proxies where lists of emergency node are stored. A proxy responds by sending back the emergent node names, estimated residual lifetimes and energy levels. The SenCar receives the message and follows the emergency recharge algorithm to recharge those nodes.

When a head node is low on energy, it can choose another node with high energy, and send out a *head notification* message to notify the latter to become the new head.

B. Protocol Design

We describe the detailed protocol assuming the head hierarchy has l levels.

1) Head Selection: After the areas and names have been configured, the network performs head selection from the bottom up starting at the l-th level. Since initially sensor nodes have

about the same level of energy, any of them may become a head. Each sensor node i generates a random probability x. If $x > K$, where K is a pre-determined threshold, the node floods a *head selection* message in its l-th level subarea, containing the name of this subarea, $x_{max} = x$, and ID_{max} set to its own ID. Otherwise the node waits for messages from other nodes in the area.

A node receiving such a *head selection* message compares the x_{max} in the message with its local record x_{max}. If its local record is larger, the message is discarded. Otherwise, the sensor updates its local x_{max} to that in the message, sets ID_{max} to that in the message, and forwards the message to its neighbors except the one that sent it this message. Finally the node with the maximum x wins the election and is recorded by all the nodes in this subarea as the head.

New heads at the l-th level then contend to become heads of the $(l-1)$-th level. They flood new *head selection* messages in the $(l-1)$-th level subarea, carrying the area's name, their respective x values and IDs. Intermediate nodes perform similar comparisons. This will elect the heads at the $(l-1)$-th level. This process is repeated until head nodes of all levels are elected.

One difference for the *head selection* messages starting from the $(l-1)$-th level and up is that messages carrying smaller x than the local copy are not discarded. Instead, they are propagated throughout the respective subarea. This builds routing states in intermediate nodes of the subarea: An intermediate node has one entry for each child head, pointing to the neighbor from which the message from that head arrives first. Duplicate copies of the same message arriving later are discarded.

Such states are effectively FIB (Forward Interest Base) entries in NDN. Later a parent head at the $(k-1)$th-level can send *energy interest* messages to its child heads at the k-th level using such states. To build FIB entries for the 1st level head nodes, they each flood a top level *head selection* message throughout the whole network. Later the energy interest queries from the SenCar can use such states to reach them.

2) Normal Energy Interest Propagation: After the head hierarchy is constructed, the SenCar sends *energy interest* messages to query for nodes needing recharge. The energy information is gathered on demand, and top down in the hierarchy. We will describe normal energy information collection first. Emergency information is collected similarly, but with only top level heads involved to reduce latency.

For normal energy information, an interest message is sent by the SenCar (e.g., with data name set to "/energy/normal/*", the energy information at all top level heads). Intermediate nodes use the FIB entries established by top level *head selection* messages to forward it to all top level head nodes. To guide the return of future data, an intermediate node also sets up a PIT (Pending Interest Table) entry pointing to the neighbor from which the interest message is received. Later energy information from a head can follow such directions to return to the SenCar.

Upon receiving an energy interest message, a first level head sends a new energy interest message to its child heads, with the data name set to all subareas of its children (e.g., from the head node of area /a, "/energy/normal/a/*"). Similarly, these messages reach all child heads following FIB entries. Intermediate nodes also set up PIT entries so later energy information from child heads can go back to their parent head. This process is repeated down the hierarchy, until finally heads at bottom level flood their respective subareas with interest messages.

3) Normal Energy Report and Node Recharge: When a sensor node receives a l-th level *energy interest* message, it responds with an *energy* message including its ID and residual energy. With the help of PIT entries, the message is returned to the head of the l-th level.

The head examines if the reported residual energy is less than the normal recharge threshold. If so, the ID of the node is added to a list, and the energy that can be recharged to this node is added to a summation counter. After the head has collected these messages, it sends an aggregation message, containing the list, the summation counter and its subarea name to its parent head. A parent head compares such messages from its child heads, selects the one with the largest summation counter (i.e., the bottom level subarea that can be recharged of the greatest amount of energy), and forwards to its parent head. This process is repeated upwards in the hierarchy. Finally, the SenCar receives one message from each top level head. The SenCar then moves to the bottom level subarea with the largest summation counter, and recharge those nodes in the ID list one by one. Only after recharging those nodes will the SenCar send another normal energy query.

The reason we delegate selection partially to head nodes is twofold. First, we expect much less variation in normal energy levels. Thus the SenCar can choose one bottom level subarea and finish recharging all listed nodes. Only after the whole subarea is recharged, we expect enough changes in normal energy distribution that warrants a new normal energy query from the SenCar. Thus it is not necessary to include the energy information from other sensors because that will be collected again. Second, this also keeps the return message sizes small and reduces overhead.

4) Emergency Energy Report and Node Recharge: Emergency energy report is slightly different due to the urgency. Each node periodically examines its energy level. If the level is below the emergency recharge threshold, it immediately sends an *emergency* message containing its ID and energy level to its proxy (i.e., its top level head node). Because the head node floods a top level *head selection* message during head election, the same FIB entries can be used to forward emergency messages to the head.

Instead of waiting for recharging a whole bottom level subarea, the SenCar sends out an *emergency interest* message to each proxy (i.e., the top level head nodes) after finishing recharging any single normal or emergency node. The proxies return their lists of emergency node IDs and energy levels, if there exists any. The SenCar interrupts its normal recharge, switches to emergency operation mode and recharge those nodes first. It switches back to normal operation mode when no more emergency is reported.

5) Head Hierarchy Maintenance: A head can be short on energy, which can happen once in a while because the head usually engages in more activities than a normal node. When this happens, a new head is needed. Because only heads of bottom levels contend for higher level elections, a head at any level is always the head of its bottom level subarea. It receives the energy reports from normal nodes in its bottom level subarea upon the normal interest query from the SenCar. So it can choose a node with the highest energy, and floods a *head notification* message to notify all nodes in the bottom level subarea of the new head.

The new head then triggers a new head election process in its $(l-1)$-th level subarea. It propagates a new *head selection* message in its $(l-1)$th subarea, but carrying its energy level instead of the random number x. Other heads in this $(l-1)$-th level subarea do the same. Then a new $(l-l)$-th head with the maximum energy is elected. If this is the same head, the process stops. Otherwise, the new $(l-1)$-th level head triggers the same process in its upper level subarea, until finally a new top level head is elected.

C. Summary of Protocol Design

We now summarize how we use NDN to route different messages briefly. First, FIB entries are established during the head selection process so that the *interest* message can be sent from parent head nodes to child ones. Second, the propagation of *interest* messages from the SenCar, or from parent to child heads, establishes PIT entries for later return of *energy* messages from top level heads or child heads. Third, FIB entries to top level heads (i.e., proxies) allow emergency nodes to send reports to proxies without waiting for the emergency interest queries from the SenCar, which minimizes latency.

V. Theoretical Analysis

We investigate a couple important theoretical questions in this section. First, what is the condition for the network to operate perpetually? Second, when multiple emergencies occur, what is the optimal recharging policy such that no sensor node would exhaust energy and the total energy replenished into the network is maximized?

A. Energy Neutral Condition

A rechargeable sensor network achieves perpetual operation when the *energy neutral* condition holds, i.e., for each sensor node the energy provided is no less than the energy consumed in any arbitrarily large time period.

We assume that the energy consumption in each time slot on a sensor is a random variable. A long time period T_n is equally slotted into n slots. Let R_{T_n} and E_{T_n} denote the energy replenished and consumed for a sensor node during time period T_n, respectively, and E_0 denote the node's initial energy. Thus the energy neutral condition is:

$$R_{T_n} + E_0 \geq E_{T_n} \tag{1}$$

We first estimate a loose upper bound for R_{T_n}. Intuitively, a SenCar reaches its maximum recharging capacity when it can "barely" keep up with the recharging needs. This is when it keeps recharging node after node without any idle time in between, and each node has almost zero energy before being recharged.

The SenCar can replenish at most the battery's full capacity in the full recharge time (We assume fully recharging batteries to avoid "memory effects" that can reduce the number of chargeable cycles). Thus the average recharge power R (i.e., the rate energy is replenished) for the whole network is the full battery capacity divided by the full recharge time (e.g., 780 mAh divided by 80 min for a Panasonic Ni-MH AAA battery [11]). The average recharge power r for each sensor node is R further divided by N, the number of nodes. Thus, the upper bound of $R_{T_n} = rT_n$.

E_{T_n} is a random variable and its probability distribution can be estimated using the central limit theorem. We have the following lemma.

Lemma 1. *The distribution of the energy consumption E_{T_n} of a sensor node in a long time period T_n asymptotically converges to a Normal Distribution.*

Proof: We model the energy consumption of a sensor node in a unit time slot as an independent and identically distributed random variables X_1 with mean $\mu(1)$ and standard deviation $\sigma(1)$. Using the Central Limit Theorem, the summation of many such i.i.d. random variables follow a Normal distribution of $\mu(n) = n\mu(1)$ and $\sigma^2(n) = n\sigma^2(1)$. ■

As a result, when the time period T_n is long enough, $E_{T_n} \sim \mathcal{N}(\mu(n), \sigma^2(n))$. Therefore, the energy neutral condition on a sensor node holds with the following probability:

$$p = Pr\{R_{T_n} + E_0 > E_{T_n}\} = \Phi\left(\frac{R_{T_n} + E_0 - \mu(n)}{\sqrt{\sigma^2(n)}}\right) \tag{2}$$

Given an application, p is a constant and can be calculated in Eq. 2. The probability to have k out of N sensor nodes working in a network follows a Binomial distribution (i.e., $k \sim \mathrm{B}(N, p)$).

To illustrate how Eq. (2) can be used to estimate network longevity, we consider a concrete example comprised of 800 sensor nodes with the recharge function of Panasonic Ni-MH AAA batteries. For a time period T_n of 6 months and each time slot is 1 second, the average total energy recharged R_{T_n} for a sensor node is 76.18 KJ. Assume in an application the battery can last 32 days on average without recharge and the standard deviation is 1.5 times of the mean, which yields $\mu(1) = 5.9mJ$ and $\sigma^2(1) = 77mJ^2$. Using Eq. (2), we can calculate $p = 0.71$. The mean value of binomial distribution is Np_i so on average we expect at most 568 nodes can work perpetually. Later in Section VI, we will see that this estimation is quite accurate, only about 10% higher than simulation results.

B. Emergency Recharge Optimization

We study the optimal recharge policy for multiple emergencies in this subsection. We consider the scenario where m emergency nodes need to be recharged and define T_{em} as the mean inter-arrival time of emergencies in a long run. The SenCar needs to recharge these nodes before they exhaust energy, and it should choose a recharging order to maximize

the amount of energy replenished into the network before a new emergency occurs (i.e., within time T_{em}).

This problem can be formulated into the famous Orienteering Problem (OP) [13]. In an OP, a set of control points associated with scores are visited by competitors before a time expiration. The competitor collecting the highest score wins the game. In analogy to the OP, the SenCar visits sensor nodes in a directed graph $G = (V, E)$, where v_1 is the starting location of the SenCar and $v_i, i \in \{2, 3, \ldots, m\}$ represents the emergent sensor locations to be visited. E is the set of routes among sensor nodes or the starting location of the SenCar. The profit r_i at v_i is the amount of energy replenished when recharging the node at this location to full capacity. The cost of traveling along e_{ij} is the traveling time from v_i to v_j (denoted as t_{ij}) plus the recharging time at v_i (denoted as t_i). To be consistent with the modeling of OP, we make the SenCar return to v_1 virtually after recharging all the selected nodes by setting $t_{i1} = 0, i \in \{2, 3, \ldots, m\}$, $t_1 = 0, r_1 = 0$. The objective is to maximize the total amount of energy refilled given the time constraint T_{em}. We introduce decision variables x_{ij} for edge e_{ij}. The decision variable is 1 if an edge is visited, otherwise it is 0. u_i is the position of vertex i in the path. The formulation of the problem is shown below.

$$\mathbf{P1}: \quad \max \sum_{i=2}^{m-1} \sum_{j=2}^{m} r_i x_{ij}, \tag{3}$$

Subject to

$$\sum_{j=2}^{m} x_{1j} = \sum_{i=1}^{m-1} x_{i1} = 1, \tag{4}$$

$$\sum_{i=1}^{m-1} x_{ik} = \sum_{j=2}^{m} x_{kj} \leq 1; \forall k = 2, 3, \ldots, m-1 \tag{5}$$

$$\sum_{i=1}^{m-1} \sum_{j=2}^{m} (t_{ij} + t_i) x_{ij} \leq T_{em}, \tag{6}$$

$$x_{ij} \in \{0, 1\}; \forall i, j = 1, 2, \ldots, m, \tag{7}$$

$$2 \leq u_i \leq m; \forall i = 2, 3, \ldots, m, \tag{8}$$

$$u_i - u_j + 1 \leq m(1 - x_{ij}); \forall i, j = 2, 3, \ldots, m \tag{9}$$

Constraint (4) guarantees that the recharge path starts at 1 and finishes at 1. Constraint (5) ensures the connectivity of the path and that every vertex is visited at most once. Constraint (6) sets the time threshold to be T_{em}. Constraint (7) imposes x_{ij} to be 0-1 valued. Constraints (8) and (9) eliminate the subtour in the planned route. The subtour elimination constraints are formulated according to [12].

There have been quite a few heuristics to solve the OP [14]–[17] and a recent survey is available in [13]. Tsiligirides [14] has developed a stochastic Monte Carlo technique and a divide-and-conquer method. Chao, et. al [16] proposed a five-step heuristic. However, they are all quite complex and a more efficient solution is desirable to find a solution in short time to avoid nodes exhausting energy. We have the following lemma.

Lemma 2. *When recharging sensor node i requires much more time than traveling from node i to node j (i.e. $t_j \gg t_{ij}$), the OP can be approximated as a Knapsack problem.*

TABLE I
ACCURACY OF KNAPSACK APPROXIMATIONS TO EXACT SOLUTIONS

# Emergencies m	3	4	5	6	7
$T_{em} = 300$ min	100%	100%	100%	100%	100%
$T_{em} = 400$ min	100%	100%	100%	100%	100%
$T_{em} = 500$ min	100%	100%	100%	100%	100%
# Emergencies m	8	9	10	11	12
$T_{em} = 300$ min	100%	100%	100%	100%	100%
$T_{em} = 400$ min	99.7%	99.6%	99.9%	99.8%	99.7%
$T_{em} = 500$ min	100%	100%	99.6%	100%	100%

Proof: Once the traveling time t_{ij} is negligible, Constraint (6) in the original OP formulation can be rewritten as $\sum_{i=2}^{m} t_i y_i \leq T_{em}$ where t_i is associated with the item weight in a Knapsack problem. The item value is the amount of energy replenished r_i. y_i is a 0-1 valued decision variable and is set to 1 only if v_i is selected for recharge. ■

Thus we have a much simpler Knapsack formulation:

$$\mathbf{P2}: \quad \max \sum_{i=2}^{m} r_i y_i, \tag{10}$$

Subject to

$$\sum_{i=2}^{m} t_i y_i \leq T_{em}. \tag{11}$$

$$y_1 = 1, \tag{12}$$

The complexity of Knapsack heuristic for m emergencies takes $\mathcal{O}(mT_{em})$ running time. It is much simpler and efficient to implement on the SenCar than the OP heuristics in [14]–[17].

To examine the accuracy of Knapsack approximation, we test several cases when the number of emergencies m varies from 3 to 12, and compare the amount of energy recharged by Knapsack approximation against the exact solution. We assume a $200 \times 200m^2$ square field, the SenCar moving at a constant speed of 3 m/s, and the recharge function follows that of Panasonic Ni-MH AAA battery. The accuracy is defined as $1 - \left|\frac{R_k - R_e}{R_e}\right|$, where R_k is the solution by Knapsack approximation and R_e is the solution by exhaustive search. Table I shows that the accuracy is more than 99% for different parameter settings.

VI. PERFORMANCE EVALUATION

In this section, we use simulation to evaluate the effectiveness and efficiency of our framework. We have developed a discrete event-driven simulator using POSIX Thread programming in C language. We use two network sizes of 500 and 800 sensor nodes, uniformly randomly distributed over a $200 \times 200m^2$ and $250 \times 250m^2$ square field respectively. The network has a 3-level hierarchy with 4^l number of areas on the l-th level. The energy consumption on each sensor is a Gaussian random variable for each time slot. The relationship between recharged energy and recharge duration follows that of Panasonic Ni-MH AAA battery.

To understand the impact of different energy consumption rates and fluctuations, we adopt two node energy consumption rates $r_c^1 \sim (4.2mW, 11mW^2)$ and $r_c^2 \sim (5.9mW, 77mW^2)$. On average, a fully recharged battery can last for 45 and 32 days at

TABLE II
PARAMETER SETTINGS

Parameter	Value
Field Length	200m, 250m
Number of Nodes N	500, 800
Number of Levels	3
Areas on l-th level	4^l
Battery Capacity	780 mAh
Transmission Rang	16 m
Consumption Rate r_c	$r_c^1 \sim (4.2mW, 11mW^2)$, $r_c^2 \sim (5.9mW, 77mW^2)$
SenCar Speed	1 m/s
Recharge Time to Full Capacity	73.4 mins
Normal Recharge Threshold	50%
Emergency Recharge Threshold	10%
Simulation Time	6 months

r_c^1 and r_c^2 respectively. The standard deviation is chosen to be 0.8 and 1.5 times the corresponding mean rate to emulate small and large fluctuations in energy variation. All the parameter settings in the simulation are listed in Table II.

A. Protocol Performance Evaluations

1) Energy Evolution and Energy Distribution: First, we show the energy evolution in the network of 500 nodes at different consumption rates. In Fig. 2, the amount of energy consumed and replenished in every one-hour time slots is plotted as functions of simulation time. Fig. 2(a) shows that the recharged energy remains at zero until after about 500 hours, then it increases steadily and stops. This is because that none of the nodes need recharge until the normal recharge threshold (50% of capacity) is reached at about 500 hours.

As sensor nodes reach that threshold, the SenCar starts normal recharge. A node recharged at a later time has consumed more energy, thereby the SenCar puts back more energy to it. This corresponds to the increase of recharged energy from about 500 to 840 hours. The recharge process pauses when the SenCar has responded to every request and resumes when a new request is received. After the network enters equilibrium, the SenCar operates in a pattern of alternating between idle and recharge as shown in Fig. 2(a). Similar behavior of the SenCar is observed in Fig. 2(b) except that the idle time is shorter because of more frequent recharge due to a higher energy consumption rate.

The energy distributions among nodes carries valuable information about the health of the network. A distribution at a higher average energy is more robust to unexpected surges in energy consumption. Fig. 3 shows the energy distribution of a 500-node network with r_c^1 and r_c^2 after the network enters energy equilibrium. In Fig. 3(a), all the nodes maintain more than 50% of the energy indicating that the SenCar has enough capacity to work only in normal operation mode and recharge all nodes. However, in Fig. 3(b), some nodes have energy under the normal recharge threshold, indicating the possibility of emergencies and the SenCar may need to switch to emergency recharge mode occasionally.

2) Number of Emergencies: Fig. 4 and Fig. 5 compare the percentage of nodes in emergency and nonfunctional (i.e., energy at zero) situations for networks of 500 and 800 nodes with different energy consumption rates. The 500 node network at consumption rate r_c^1 has no emergency since all the nodes

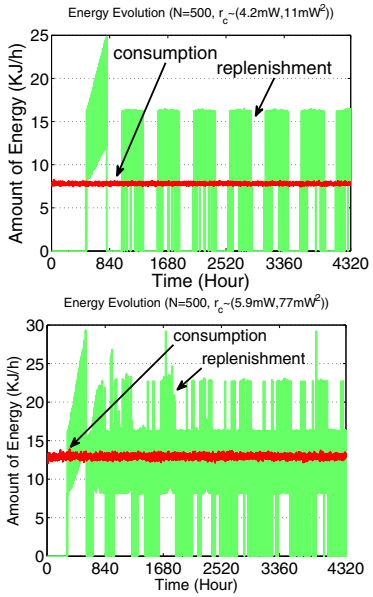

Fig. 2. Evolution of energy consumption vs. energy replenishment in 6 months time. (a) Network Size N = 500 and consumption rate $r_c \sim (4.2mW, 11mW^2)$; (b) Network Size N = 500 and consumption rate $r_c \sim (5.9mW, 77mW^2)$.

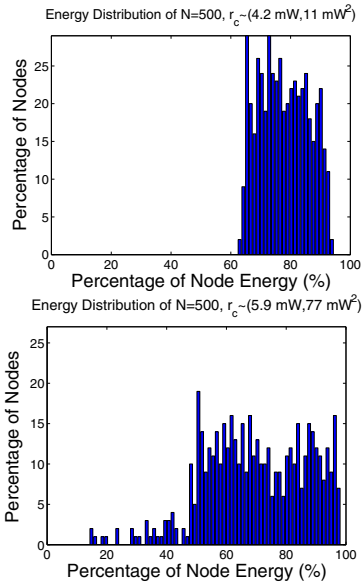

Fig. 3. Energy distribution at equilibrium. (a) Network Size N = 500 and consumption rate $r_c \sim (4.2mW, 11mW^2)$; (b) Network Size N = 500 and consumption rate $r_c \sim (5.9mW, 77mW^2)$.

are recharged timely after their energy drops below the normal recharge threshold. As the network size or energy consumption rate increases, the energy of some nodes falls below the emergency recharge threshold. When the increase goes beyond the capacity of the SenCar, it cannot recharge all of them in time and some nodes may become temporally nonfunctional.

For the 500 nodes with r_c^2 and 800 nodes with r_c^1, there are a number of nodes in emergency and they become temporally nonfunctional at the beginning. After the networks enter equilib-

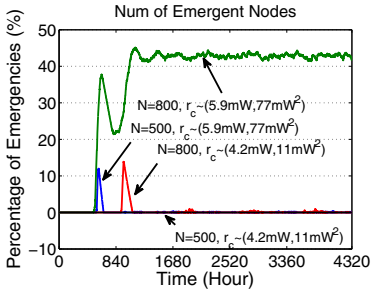

Fig. 4. Number of emergent nodes.

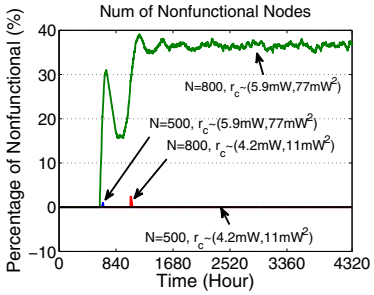

Fig. 5. Number of nonfunctional nodes.

rium, most of the nodes are recharged in time. Only a few nodes enter emergency state or become temporally nonfunctional, which occurs only sporadically. When the 800 node network has consumption rate at r_c^2, the recharging capacity of the SenCar is exceeded. Fig. 5 shows that the percentage of nonfunctional nodes holds persistently around 38%.

Now we examine how much gap exists for the maximum number of nodes the SenCar can sustain between the simulation results and the theoretical analysis. We can calculate that the 800 node network at consumption rate of r_c^2 has $p = 0.71$ probability for the energy neutral condition (Eq. (2)) to hold. Thus the SenCar can sustain at most 568 (i.e., Np) nodes. This is about 9% higher than the actual number of 496 nodes indicated by simulations.

The gap is caused mainly by two reasons. 1) The SenCar does not start recharging until there are some nodes whose energy drops below the normal recharge threshold. This idle time corresponds to the time from 0 to about 500 in Fig 4, during which no energy is refilled into the network. 2) The SenCar does not recharge nodes when it is moving or collecting energy information. Table III shows that the theoretical probability matches very well against the percentage of nonfunctional nodes from simulation results. We believe that although the energy neutral analysis gives a loose upper bound, the gap is reasonably small so network administrators can make reasonable estimations.

3) Response Time to Emergencies: We also evaluate the response time to emergencies and compare with the static optimization approaches used in [8], [9]. The response time to emergencies is measured from the time a node reports emergency until it is recharged by the SenCar. A shorter response time indicates that the SenCar can respond faster to emergencies. Fig 6 shows the SenCar's average response time to emergencies compared to the static approach when

TABLE III
PROBABILITY FOR THE ENERGY NEUTRAL CONDITION TO HOLD

N and r_c	P_{op}	%(nonfunctional)
$N = 500$, r_c^1	1.0	0
$N = 500$, r_c^2	1.0	0
$N = 800$, r_c^1	0.999	0
$N = 800$, r_c^2	0.71	38

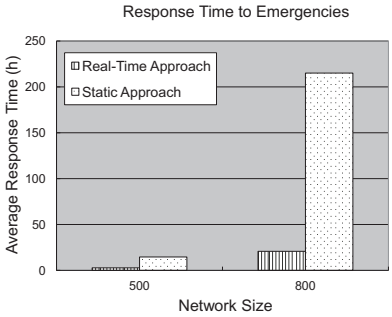

Fig. 6. Average response time to emergencies.

$N = 500, 800$ and energy consumption rate at r_c^2. In the static approach, the SenCar selects nodes with energy less than the normal recharge threshold, calculates the minimum traveling distance throughout these nodes and performs recharge one by one. We can see that when $N = 500$, the average response time in our approach is less than 3 hours whereas it requires as long as 20 hours in the static approach.

The situation becomes worse with the static approach when $N = 800$ because of the surging number of emergencies which ultimately results in as high as 56% nonfunctional nodes. The average response time increases to around 200 hours because in [8], [9] emergency nodes and normal nodes are not differentiated. A pre-computed route contains the combination of these nodes would result in extremely long waiting time for emergency nodes to get recharged. The approach becomes infeasible as the network size increases. A node in emergency would have been dead already when the SenCar arrives. In contrast, our real-time approach prioritizes nodes in emergency. The average response time is 14 hours even when $N = 800$ which is more than one order of magnitude faster than that of static approaches. It also incurs only 38% nonfunctional nodes (in Fig. 5), much less than the 56% in static approaches. Thus for extreme cases when the capacity of the SenCar is exceeded, nonfunctional situations are resolved much faster than the static approach.

B. Protocol Overhead and Cost Evaluations

1) Evaluation of Protocol Overhead: We evaluate the overhead introduced by our protocol, including all types of messages sent by sensor nodes or the SenCar to recharge nodes. Fig. 7 shows the average overhead per node in a 6 month period. The average overhead on each sensor node is from 4 to 7 bits per second, which is negligible compared to radio transmission rates in sensor nodes (e.g., 20 - 900 kbps). We also find that the overhead is slightly higher at the lower energy consumption rate, and much lower for the larger network size of 800 nodes. This can be explained by the difference between normal and emergency energy information gathering. In the 500-node network,

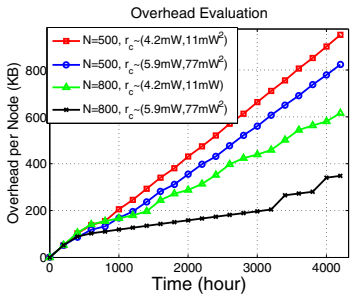

Fig. 7. Average overhead for each sensor node per second.

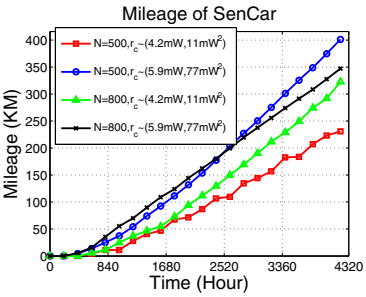

Fig. 8. Mileages SenCar travels.

the SenCar has to collect energy information and performs normal recharge operation more frequently. More messages are produced during normal recharge operation in collecting energy information along the entire head hierarchy. However, in the 800-node network, there are more emergency situations and the emergency messages are directly reported to the proxies without propagation along the head hierarchy. Thus, the overhead in the 800-node network is less than the 500-node network.

2) Evaluation of SenCar Maintenance Cost: We use the the mileages the SenCar travels to evaluate the cost (e.g., the energy consumed) for the SenCar to move around. Fig. 8 shows the accumulated mileages in 6 months. For the network with 500 nodes and consumption rates at r_c^1, there is no emergency, and it takes long time for the energy of a node to drop below the normal recharge threshold, thus the SenCar seldom moves and has the least mileages. When the energy consumption rate increases to r_c^2 for the 500-node network, the energy of nodes drops much faster. As a result, the SenCar performs normal recharge more frequently and the mileages increase significantly.

For the network with 800 nodes and the energy consumption rate set to r_c^1, emergencies occur more frequently than the network with 500 nodes at r_c^1, thus the SenCar performs emergency recharge more frequently. Since it takes longer time to recharge a node in emergency, the SenCar moves less frequently and the mileages are less than 500 nodes with consumption rate at r_c^2. When the energy consumption rate increases to r_c^2, emergencies exist persistently and the SenCar performs emergency recharge all the time, which is presented in Fig.8 as the mileages of 500-node network at r_c^2 exceed that of 800-node network at r_c^2 after 2520 hours, and the trend that the mileages of 800-node network at r_c^1 will also exceed that of 800-node network at r_c^2 shortly after 4320 hours.

C. Summary

The simulation results demonstrate that NETWRAP is very effective in achieving perpetual operation with high efficiency in wireless sensor networks. The SenCar has the capacity to maintain the network energy level and handle emergencies in large networks as no nodes deplete energy throughout the 6 month-simulation.

VII. Conclusions

In this paper, we propose a novel framework for real time wireless energy recharging for wireless sensor networks. We develop a comprehensive set of protocols using NDN concepts and mechanisms to enable effective recharging for the perpetual operation of the network. The protocols adapt to unpredictable network conditions and satisfy the needs for both normal and emergency recharging. We formally analyze the probability for the energy neutral condition required by perpetual operations. We also model the optimal recharging of multiple emergencies as an Orienteering problem and provide a Knapsack approximation that has high accuracy under typical network environments. The extensive simulation results demonstrate the efficiency and effectiveness of our framework, and the close match of energy neutral analysis with simulation results.

Acknowledgments

This work was supported in part by NSF grant numbers ECCS-0801438 and ECCS-1307576, and ARO grant number W911NF-09-1-0154.

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] A. Karalis, J. D. Joannopoulos and M. Soljacic, "Efficient wireless non-radiative mid-range energy transfer," *Annals of Physics*, vol. 323, no. 1, pp. 34-48, Jan. 2008.

[3] PowerCast Corp, "http://www.powercastco.com."

[4] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," *IEEE International Conference on Robotics and Automation*, 2003.

[5] A. Kansal, J. Hsu, M. Srivastava and V. Raqhunathan, "Harvesting aware power management for sensor networks," *43rd ACM/IEEE Design Automation Conference*, 2006.

[6] B. Tong, Z. Li, G. Wang and W. Zhang, "How wireless power charging technology affects sensor network deployment and routing", *IEEE ICDCS*, 2010.

[7] S. He, J. Chen, F. Jiang, D. Yau , G. Xing and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE INFOCOM*, 2011.

[8] M. Zhao, J. Li and Y. Yang, "Joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," *IEEE ITC*, 2011.

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

[10] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs and R. Braynard, "Networking named content," *ACM CoNEXT*, 2009.

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

[12] C. Miller, A. Tucker and R. Zemlin,"Integer programming formulations and traveling salesman problems," *Journal of the ACM*, pp. 326-329, 1960.

[13] P. Vansteenwegen, W. Souffriau and D. Van Oudheusden, "The orienteering problem: a survey," *European Journal of Operation Research*, vol. 209, no. 1, 2011.

[14] T. Tsiligirides, "Heuristic methods applied to orienteering," *Journal of the Operation Research Society*, vol. 35, no. 9, pp. 797-809, 1984.

[15] B. Golden, A. Assad and R. Dahl, "The orienteering problem," *Naval Research Logistics* 34, pp. 307-318, 1987.

[16] I. Chao, B. Golden and E. Wasil, "A fast and effective heuristic for the orienteering problem," *European Journal of Operation Research*, vol. 88, no. 3, 1996.

[17] M. Fischetti, J. S. Gonzalez and P. Toth, "Solving the orienteering problem through branch-and-cut," *INFORMS Journal on Computing*, vol. 10, no. 2, pp. 133-148, 1998.