

On Energy Provisioning and Relay Node Placement for Wireless Sensor Networks

Y. Thomas Hou, *Senior Member, IEEE*, Yi Shi, *Student Member, IEEE*,
Hanif D. Serali, and Scott F. Midkiff, *Senior Member, IEEE*

Abstract—Wireless sensor networks that operate on batteries have limited network lifetime. There have been extensive recent research efforts on how to design protocols and algorithms to prolong network lifetime. However, due to energy constraint, even under the most efficient protocols and algorithms, the network lifetime may still be unable to meet the mission's requirements. In this paper, we consider the energy provisioning (EP) problem for a two-tiered wireless sensor network. In addition to provisioning additional energy on the existing nodes, we also consider deploying relay nodes (RNs) into the network to mitigate network geometric deficiencies and prolong network lifetime. We formulate the joint problem of EP and RN placement (EP-RNP) into a mixed-integer nonlinear programming (MINLP) problem. Since an MINLP problem is NP-hard in general, and even state-of-the-art software and techniques are unable to offer satisfactory solutions, we develop a heuristic algorithm, called Smart Pairing and Intelligent Disc Search (SPINDS), to address this problem. We show a number of novel algorithmic design techniques in the design of SPINDS that effectively transform a complex MINLP problem into a linear programming (LP) problem without losing critical points in its search space. Through numerical results, we show that SPINDS offers a very attractive solution and some important insights to the EP-RNP problem.

Index Terms—Energy provisioning, flow routing, network lifetime, power control, relay node placement, wireless sensor networks.

I. INTRODUCTION

WIRELESS sensor networks have attracted unprecedented attention in recent years. In this paper, we consider a two-tiered wireless sensor network that can be used for a wide range of applications. Under the two-tiered architecture, a wireless sensor network consists of a number of sensor clusters and a base station (BS). Each cluster is deployed around a strategic location and consists of a number of microsensor nodes (MSNs) and one aggregation and forwarding node (AFN). Each MSN is used to capture and transmit collected information to the local AFN, and the AFN performs in-network processing by aggregating all collected local information. The

Manuscript received March 4, 2004; revised August 30, 2004; accepted November 7, 2004. The editor coordinating the review of this paper and approving it for publication is W. Liao. This work has been supported in part by the National Science Foundation under Grants ANI-0312655, CNS-0347390, and DMI-0094462, and by the Office of Naval Research under Grant N00014-03-1-0521.

Y. T. Hou, Y. Shi, and S. F. Midkiff are with The Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA (e-mail: thou@vt.edu; yshi@vt.edu; midkiff@vt.edu).

H. D. Serali is with the Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: hanfisd@vt.edu).
Digital Object Identifier 10.1109/TWC.2005.853969

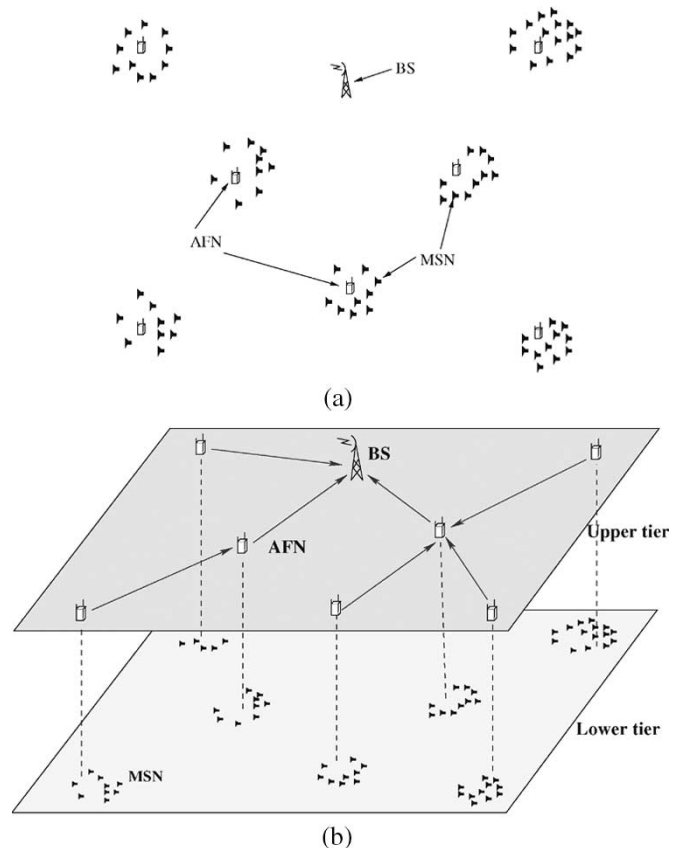


Fig. 1. Reference architecture for a two-tiered wireless sensor network. (a) Physical topology. (b) Hierarchical view.

AFN then relays the composite information to the BS via a single or multihop transmission (see Fig. 1).

An important performance measure for wireless sensor networks is the network lifetime. Recent researches on maximizing network lifetime focus on devising optimal flow routing algorithms (see, e.g., [4]) based on power control of each node's transmitter. However, the network lifetime, even under optimal flow routing, may still not be able to meet the mission's requirements, and consequently, other strategies are needed.

For the two-tiered wireless sensor network, although MSNs are not expected to be reprovisioned with additional energy due to their small size, low cost, and large number,¹ it is plausible to consider provisioning energy to the upper tier AFNs if we wish to extend the network's lifetime (see Section II for more

¹It is expected that new MSNs can be redeployed in the same area to replace those MSNs that have either failed or run out of energy.

details). Further, limiting energy provisioning (EP) only to the existing AFNs may not yield the most efficient solution. This is because node energy consumption behavior and network lifetime performance in a wireless sensor network are highly dependent on the network geometry. As we shall see later in this paper, it is more efficient to deploy additional relay nodes (RNs) in the network to mitigate certain network geometric deficiencies.

In this paper, instead of studying EP (on existing AFNs only) and RN placement (RNP) problems separately, we investigate the joint problem of EP and RNP for sensor networks. We also generalize the notion of EP in the sense that energy can be provisioned on either AFNs or RNs. As a result, our work can be applied to address a wide range of problems associated with EP or RNP.

Specifically, we investigate the following problem for EP–RNP: for a given network and some initial energy at each AFN, how should we allocate a total amount of additional energy E at M locations (which can be either at an existing AFN or at a new position for RN) such that the network lifetime can be maximized? We show that this EP–RNP problem can be cast into a mixed-integer nonlinear programming (MINLP) problem. Since an MINLP problem is known to be NP-hard, and even state-of-the-art techniques (e.g., branch-and-bound [9]) and their software implementations (e.g., BARON [2]) cannot provide a good solution, we resort to develop an efficient heuristic algorithm.

Our heuristic algorithm is called SPINDS, which stands for Smart Pairing and INtelligent Disc Search. SPINDS is an iterative algorithm that attempts to increase the network lifetime by iteratively moving an RN to a better location. Our main idea in achieving this objective is to transform the original MINLP problem into a linear programming (LP) approximation. This is achieved by two ingenious steps. In the first step, we use the so-called smart pairing (SP) and intelligent disc search (INDS) techniques to determine possible RN placements during each iteration such that network lifetime can be increased. This step transforms the original MINLP problem into a mixed-integer linear programming (MILP) problem. Although the MILP problem appears simpler than the MINLP problem, it is still NP-complete in general. In the second step, we introduce an equivalence lemma, which shows that if the RNs are placed wisely, then the MILP problem could be sub-breakstituted by a much simpler LP problem without any compromise in network lifetime performance. Consequently, it is possible to transform the original MINLP problem into an iterative LP problem, which is polynomial.

In the numerical results, we show that the proposed SPINDS can indeed place the RNs wisely and the LP substitution indeed matches the MILP formulation. We also show that SPINDS offers highly competitive performance in solving the EP–RNP problem when compared with some other approaches. Furthermore, we offer some important insights on network geometric properties, RN placement, and EP. We show that deficiencies due to network geometry (or topology) have a significant impact on network lifetime. When such deficiencies exist, RN placements can be a much more efficient technique than merely provisioning additional energy on existing AFNs.

The remainder of this paper is organized as follows. Section II describes the two-tiered wireless sensor network architecture and gives some background on power consumption and power control. We also formulate EP–RNP as an MINLP problem. Section III presents SPINDS, a polynomial-time algorithm to solve the joint EP–RNP problem. Section IV uses numerical results to demonstrate the efficacy of SPINDS and offers insights on network geometry, EP, and RNP. Section V discusses related work and Section VI concludes this paper.

II. SYSTEM MODELING AND PROBLEM FORMULATION

A. Reference Network Architecture

We focus on a two-tiered architecture for wireless sensor networks. The two-tiered network architecture is motivated by recent advances in distributed source coding (DSC) [5], [17], [21], which can exploit redundancy in information collected among neighboring sensors without intersensor communications (Slepian–Wolf [23] and Wyner–Ziv [26] theorems).

Fig. 1(a) and (b), respectively, shows the physical and hierarchical network topology for such a network. As shown in the figures, we have three types of nodes in the network: MSNs, AFNs, and a BS. MSNs can be application-specific sensor nodes and constitute the lower tier of the network. They are deployed in groups (or clusters) at a strategic location for surveillance or monitoring applications. Each MSN is small and low cost; they are densely deployed within a small geographic area. The objective of an MSN is very simple: once triggered by an event, it starts to capture live information, which it sends directly to the local AFN in one hop. It is worth pointing out that multihop routing among MSNs is not necessary due to the small distance between an MSN and its AFN. Moreover, an MSN will cease to function once its battery runs out of energy. By deploying these inexpensive MSNs in clusters and within proximity of a strategic location, it is possible to obtain a comprehensive view of the area situation by exploring the correlation among the scenes collected at each MSN [5]. Furthermore, the reliability of area surveillance capability can also be improved through redundancy among the MSNs in the same cluster.

For each cluster of MSNs, there is one AFN that is different from an MSN in terms of physical properties and functions. The primary functions of an AFN are 1) data aggregation (or “fusion”) for data flows coming from the local cluster of MSNs and 2) forwarding (or relaying) the aggregated information to the next hop AFN toward the BS. For data fusion, an AFN analyzes the content of each data stream it receives, from which it composes a complete scene by exploiting the correlation among each individual data stream from the MSNs. An AFN can also serve as an RN for other AFNs to carry traffic toward the BS. Although an AFN is expected to be provisioned with much more energy than an MSN, it also consumes energy at a substantially higher rate (due to wireless communication over large distances). Consequently, an AFN has limited lifetime. Upon the depletion of energy at an AFN, we expect that the coverage for the particular area under surveillance will be lost despite the fact that some of

TABLE I
 NOTATION

Symbol	Definition
N	Total number of AFNs in the network
T	Network lifetime
E	Total provisioning energy
M	Total number of energy provisioning locations
e_i	Initial energy at AFN i
g_i	Local generated rate at AFN i
ρ	Power consumption coefficient for receiving data
α	Distance independent term in power consumption coefficient for receiving data
β	Distance dependent term in power consumption coefficient for receiving data
m	Path loss index
c_{ik}	Power consumption coefficient for transmitting data from AFN i to AFN k (or the base-station B)
(or c_{iB})	
d_{ik}	Distance from AFN i to AFN k
(or d_{iB})	(or the base-station B)
D_{ik}	Distance from AFN i to AFN k
(or D_{iB})	(or the base-station B)
f_{ik}	Rate from AFN i to AFN k
(or f_{iB})	(or the base-station B)
V_{ik}	Total volume from AFN i to AFN k
(or V_{iB})	(or the base-station B)
λ_i	If we provision energy at location i , $\lambda_i = 1$; otherwise, $\lambda_i = 0$
μ_i	At location i , we provision μ_i of total energy
L_{\max}	maximum distance from an AFN to base-station

the MSNs within the cluster may still have some remaining energy.² Therefore, the most stringent definition for network lifetime would be the time instance when any one of the AFNs fails. We will use this definition throughout this paper.

The last component in the two-tiered architecture is the BS. The BS is, essentially, the sink node for data streams from all the AFNs in the network. A BS may be assumed to have a sufficient battery resource provision, or its battery may be reprovioned during its course of operation. Therefore, its power dissipation is not a concern in our investigation.

In summary, the main function of the lower tier MSNs is data acquisition and compression while the upper tier AFNs are used for data fusion and relaying the information to the BS. The routing topology can be controlled by the power level of the transmitter ([18], [20], [25]) because it directly controls the distance coverage of an AFN. That is, by adjusting the power level of the transmitter, the authors can form different network routing topologies with different network lifetime performance.

B. Power Dissipation

Table I lists the notation used in this paper. For ease of exposition, we assume that the rate of data stream g_i generated at AFN i (after data aggregation) is of constant bit rate.³ For an AFN, the power consumption by data communication (i.e., receiving and transmitting) is the dominant factor [1]. The power dissipation at the transmitter can be modeled as

$$p_t(i, k) = c_{ik} \cdot f_{ik} \quad (1)$$

²We assume that each MSN can only forward information to its local AFN for processing.

³The approach presented in this paper can be extended for the case when g_i is time varying, as long as the average of g_i is known *a priori* [11].

where $p_t(i, k)$ is the power dissipated at node i when it is transmitting to node k , f_{ik} is the bit rate transmitted from node i to k , and c_{ik} is the power consumption cost of radio link (i, k) and is given by

$$c_{ik} = \alpha + \beta \cdot d_{ik}^m \quad (2)$$

where α is a distance-independent constant term, β is a coefficient term associated with the distance-dependent term, d_{ik} is the distance between these two nodes, and m is the path loss index, with $2 \leq m \leq 4$ [19]. Typical values for these parameters are $\alpha = 50$ nJ/bit and $\beta = 0.0013$ pJ/bit/m⁴ (for $m = 4$) [10].⁴ The power dissipation at a receiver can be modeled as [19]

$$p_r(i) = \rho \cdot \sum_{k \neq i} f_{ki} \quad (3)$$

where $\sum_{k \neq i} f_{ki}$ [in bit per second (b/s)] is the rate of the received data stream (from other AFNs) at node i . A typical value for the parameter ρ is 50 nJ/bit [10].

C. The Joint EP and RNP Problem

For a network with N AFNs, where each AFN i generates data with rate g_i , suppose that the initial energy at each node is e_i ($1 \leq i \leq N$). Then, it is straightforward to use an LP approach to find an optimal flow routing schedule such that the network lifetime is maximized [4].

Now, we take one step further. Suppose that for a number of reasons this network lifetime is not adequate to meet the required lifetime. Then, it is necessary to take some measures to prolong the network lifetime. One straightforward measure is to provision additional energy on existing AFNs in the network. As we shall see later in this paper, there may exist intrinsic geometric deficiencies within the underlying network topology that cannot be efficiently addressed by just adding more energy on existing AFNs. Instead, a powerful technique to mitigate such geometric deficiencies would be to deploy additional RNs at certain locations into the network (see Fig. 2). Physically, these RNs are very much similar to AFNs, except that they do not generate any information locally as AFNs; RNs are used solely to relay network traffic toward the BS. We will show that deploying these RNs at certain critical positions in the network is much more efficient than just adding the same amount of energy on existing AFNs.

For a given pool of energy E and M RNs, the question to ask becomes: where should we deploy RNs into the network and how should we allocate the total amount of energy E into M portions such that the network lifetime can be maximized?⁵

There is one subtle problem that needs to be clarified. Should we find that an RN happens to coincide with an AFN,

⁴In this paper, we use $m = 4$ in all of our numerical results.

⁵In this paper, we assume that energy provisioning can be done with arbitrary precision (i.e., a real number). This corresponds to the case where battery could be recharged instead of replaced. In the case when battery can only be replaced, we can quantize the real value result from this paper to obtain an integer solution.

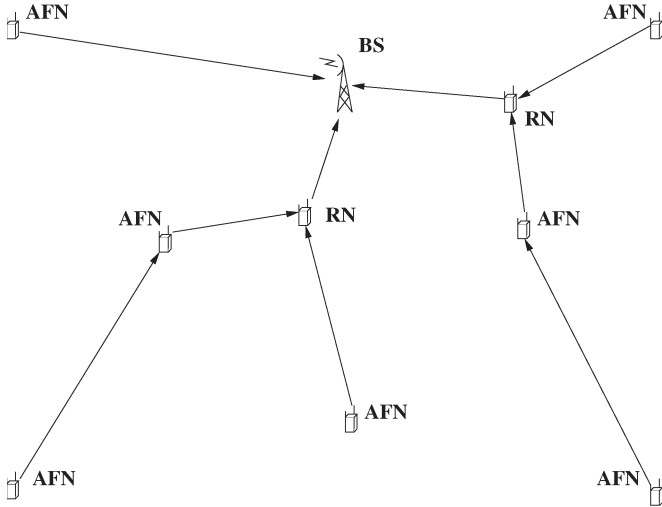


Fig. 2. Traffic routing with RNs.

what should we do with this RN? In this case, there is really no need to deploy this additional RN since we can provision the same amount of additional energy directly onto an existing AFN while achieving the same effect. Under this setting, a general interpretation for the number M might be that it represents the maximum number of possible locations that we can provision energy into the network.

For the joint EP-RNP problem, assume that the data rates from node i to node k and to the BS B are f_{ik} and f_{iB} ; (x_i, y_i) , $N < i \leq N + M$, are variable coordinates for the placements of the M RNs; d_{ik} and d_{iB} are the distances from node i to node k and to the BS B ; and c_{ik} and c_{iB} are the link costs from node i to node k and to the BS B , respectively. For each AFN i , $1 \leq i \leq N$, the following flow balance equation and energy constraint must be met:

$$f_{iB} + \sum_{1 \leq k \leq N+M, k \neq i} f_{ik} = \sum_{1 \leq m \leq N+M, m \neq i} f_{mi} + g_i,$$

$$\sum_{1 \leq k \leq N} c_{ik} f_{ik} T + \sum_{N+1 \leq k \leq N+M} (\alpha + \beta d_{ik}^4) f_{ik} T$$

$$+ \sum_{1 \leq m \leq N+M, m \neq i} \rho f_{mi} T + c_{iB} f_{iB} T - E \mu_i \leq e_i.$$

For each RN i , $N < i \leq N + M$, it must also meet the flow balance equation and energy constraint

$$f_{iB} + \sum_{1 \leq k \leq N+M, k \neq i} f_{ik} = \sum_{1 \leq m \leq N+M, m \neq i} f_{mi},$$

$$\sum_{1 \leq m \leq N+M, m \neq i} \rho f_{mi} T + \sum_{1 \leq k \leq N+M, k \neq i} (\alpha + \beta d_{ik}^4) f_{ik} T$$

$$+ (\alpha + \beta d_{iB}^4) f_{iB} T - E \mu_i \leq 0$$

where $\sum_{i=1}^{N+M} \mu_i = 1$ and at most M of them are positive.

Denote $V_{ik} = f_{ik} T$, $V_{iB} = f_{iB} T$, $D_{ik} = d_{ik}^4$, and $D_{iB} = d_{iB}^4$, we formulate the EP-RNP problem as follows.

(EP-RNP) Maximize T subject to

$$V_{iB} + \sum_{1 \leq k \leq N+M, k \neq i} V_{ik} - \sum_{1 \leq m \leq N+M, m \neq i} V_{mi} - g_i T = 0$$

$$(1 \leq i \leq N) \quad (4)$$

$$V_{iB} + \sum_{1 \leq k \leq N+M, k \neq i} V_{ik} - \sum_{1 \leq m \leq N+M, m \neq i} V_{mi} = 0$$

$$(N < i \leq N + M) \quad (5)$$

$$\sum_{1 \leq k \leq N, k \neq i} c_{ik} V_{ik} + \sum_{N+1 \leq k \leq N+M} (\alpha V_{ik} + \beta D_{ik} V_{ik})$$

$$+ \sum_{1 \leq m \leq N+M, m \neq i} \rho V_{mi} + c_{iB} V_{iB} - E \mu_i \leq e_i \quad (1 \leq i \leq N) \quad (6)$$

$$\sum_{1 \leq m \leq N+M, m \neq i} \rho V_{mi} + \sum_{1 \leq k \leq N+M, k \neq i} (\alpha V_{ik} + \beta D_{ik} V_{ik})$$

$$+ \alpha V_{iB} + \beta D_{iB} V_{iB} - E \mu_i \leq 0 \quad (N < i \leq N + M) \quad (7)$$

$$\sum_{i=1}^{N+M} \lambda_i = M \quad (8)$$

$$\sum_{i=1}^{N+M} \mu_i = 1 \quad (9)$$

$$\mu_i - \lambda_i \leq 0 \quad (1 \leq i \leq N+M) \quad (10)$$

$$(x_i - x_k)^2 + (y_i - y_k)^2 - D_{ik}^{0.5} \leq 0 \quad (1 \leq i, k \leq N+M, i \text{ or } k > N, i \neq k) \quad (11)$$

$$(x_i - x_B)^2 + (y_i - y_B)^2 - D_{iB}^{0.5} \leq 0 \quad (N < i \leq N+M) \quad (12)$$

$$c_{ik} = \alpha + \beta D_{ik} \quad (1 \leq i, k \leq N+M, i \text{ or } k > N, i \neq k) \quad (13)$$

$$c_{iB} = \alpha + \beta D_{iB} \quad (N < i \leq N+M) \quad (14)$$

$$V_{ik}, V_{iB}, D_{ik}, D_{iB}, c_{ik}, c_{iB} \geq 0 \quad (1 \leq i, k \leq N+M, i \neq k)$$

$$T, \mu_i \geq 0, \lambda_i = 0 \text{ or } 1 \quad (1 \leq i \leq N+M). \quad (15)$$

The physical interpretation of the above formulation is as follows. The constraints in (4) are bit volume balance equations for those AFNs that can generate their own traffic. The constraints in (5) are bit volume balance equations for those potential RNs (that do not generate their own traffic). The inequalities in (6) are the energy constraints for the N AFNs. The inequalities in (7) are the energy constraints for the M potential RNs. The first term in these inequalities represents

the energy spent on receiving data streams from other nodes, and the second term represents the energy spent on transmitting data streams to other nodes. The constraints in (8)–(10), and (15) ensure that we can provision energy to at most M locations (including stand-alone RNs and those RNs that coincide with AFNs). In particular, the constraints in (10) and (15) assert that $0 \leq \mu_i \leq 1$, λ_i can be only 0 or 1, and $\mu_i > 0$ only if $\lambda_i = 1$. The constraints in (11) and (12) represent the internodal distances, whereas the constraints in (13) and (14) represent the link costs among the nodes.⁶

III. SPINDS: A COMPETITIVE HEURISTIC SOLUTION

The problem formulation for an EP–RNP is in the form of a mixed-integer nonlinear programming (MINLP) problem, which is NP-hard in general [7]. The state-of-the-art techniques for solving the MINLP problem include generalized benders decomposition [8], outer approximation [6], and branch-and-bound [9] methods. Since our problem is nonconvex, the generalized benders decomposition and outer approximation methods would not work well. The current state-of-the-art software for solving this type of problem is BARON [2], which was developed by Prof. N. Sahinidis’ group at the University of Illinois and is based on branch-and-bound/reduce techniques [24]. For EP–RNP, we find that BARON can only give a reasonably good solution when N and M are very small (e.g., N less than 5), and it fails to provide reasonable lower and upper bounds for a network of moderate size.

In this section, we present a competitive heuristic algorithm to the EP–RNP problem. Before we describe this algorithm, we present some basic results on the relationship among the search space, MILP/LP formulation, and optimality, which will be the basis for some of the simplifications we will make in the algorithmic design.

A. Some Basic Results

Suppose that the locations for the M potential RNs are fixed (although they may not be optimally placed). Then, c_{ik} and c_{iB} [in (13) and (14)] are now all constants and we can solve the EP problem by tackling an MILP problem, which we call EP(AFN+RN). EP(AFN+RN) attempts to allocate a total amount of additional energy E to M points, where these M points are an optimal set of M nodes drawn from the collection of N AFNs and M RNs. Unfortunately, an MILP problem is NP-complete in general [7]. Although there exist softwares (e.g., CPLEX and LINDO) for solving MILP problems, the computational time with such softwares is only acceptable for a one-time computation. In other words, such softwares are not suitable for a large number of repetitive routine calls as required in our heuristic algorithm. To ensure that the heuristic algorithm is computationally efficient, we must find an alternative approach other than solving the MILP problem directly.

⁶Note that we use inequalities in (11) and (12) instead of equalities. This is because these inequalities have the convexity property and thus are easier to solve. After we obtain an optimal solution, we can change these inequalities back to equalities by reassigning exact values to D_{ik} ’s and D_{iB} ’s. Consequently, we will obtain an optimal solution [3].

Let us examine the following simplified problem. Instead of drawing an optimal set of M points out of the N AFNs and M RNs, we consider provisioning energy only to the M RNs and denote this problem as EP(RN). The problem formulation for EP(RN) is similar to the EP(AFN+RN) problem except that the sets of constraints $\sum_{i=1}^{N+M} \mu_i = 1$, $\sum_{i=1}^{N+M} \lambda_i = M$ ($\lambda_i = 0$ or 1), and $\mu_i - \lambda_i \leq 0$ are replaced by $\sum_{i=N+1}^{N+M} \mu_i = 1$ and $\mu_i = 0$ ($1 \leq i \leq N$); and when an RN coincides with an AFN, there is no energy consumption for receiving and transmitting a data stream between them. Clearly, this EP(RN) problem is an LP and can be solved efficiently [13].

It is not hard to see that for the same fixed network topology and initial energy on each AFN, an optimal solution for the EP(RN) problem is not better than an optimal solution for the EP(AFN+RN) problem. This is intuitive and can be easily proved by noting that the solution for the EP(RN) only considers one special case for the EP(AFN+RN), i.e., provisioning energy only to the M RNs.

Now, let us consider the following situation. Instead of comparing the solutions to EP(AFN+RN) and EP(RN) for one topology instance, how about that we try out all possible locations for placing the M RNs and compare the best solution under EP(AFN+RN) and EP(RN) among all possible placement topologies? The answer to this question is a key to our algorithmic design and is given in the following lemma.

Lemma 1 (Optimal Equivalence): Suppose that the M RNs can be arbitrarily placed over some region (including those locations for the N AFNs). Then, the best placement solution (among all possible solutions) for problem EP(AFN+RN) yields the same network lifetime performance as the best placement solution for problem EP(RN).

Proof:

- 1) It is not hard to see that for the same fixed network topology and initial energy on each AFN, the optimal solution for the EP(RN) problem is not better than the optimal solution for the EP(AFN+RN) problem. Consequently, the best placement solution (among all possible solutions) for problem EP(RN) is not better than the best placement solution for problem EP(AFN+RN).
- 2) We now show the converse is also true. For the EP(AFN+RN) problem, we have $\binom{N+M}{M}$ possible strategies of adding energy to M points. But each strategy corresponds to one instance of placement strategy for the EP(RN) problem. Therefore, any optimal solution for this EP(AFN+RN) problem has also been considered by the EP(RN) problem under a certain instance of placement strategy. Hence, the best solution to the EP(AFN+RN) problem among all possible RN placement topologies is not better than the best solution to the EP(RN) problem among all possible RN placement strategies. The proof is now complete. ■

Lemma 1 suggests that if we choose the M -node placement points wisely, then the solution to the simpler EP(RN) problem will yield a similar result as that to the EP(AFN+RN) problem. We will exploit this result in the design of our heuristic algorithm. Fig. 3 shows the relationship among all the problems we have explored so far in this paper. EP–RNP is an

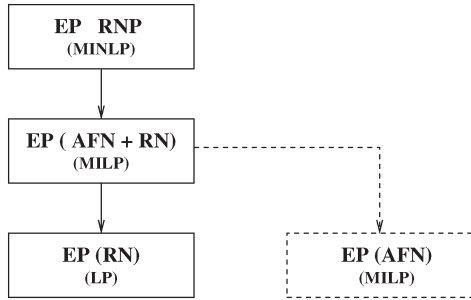


Fig. 3. Relationship among the problems and their complexity in our investigation.

MINLP problem, which is NP-hard and most difficult to solve. If we assume that the locations for the M RNs are fixed, then the EP-RNP problem becomes EP(AFN+RN), which is an MILP problem and is NP-complete in general. However, for a one-time computation, software packages such as LINDO can solve it in an acceptable time for the sizes of network under our investigation. On the other hand, if we consider provisioning energy only onto the RNs, then the EP(AFN+RN) problem becomes EP(RN), which is LP and can be solved efficiently. In Section IV, we will also consider the case that energy is only added onto the existing AFNs (without RNs). This makes the EP(AFN+RN) problem become EP(AFN), which is an MILP problem.

B. SPINDS: Procedural Description

We are now ready to present our heuristic algorithm. The heuristic algorithm that we developed is called SPINDS. The main idea of SPINDS is as follows. Suppose we start with some initial locations for the M RNs. If these locations for M RNs are not optimal, then it is possible to relocate some RNs to a better location so that the network lifetime can be further extended. Now, we repeat this process iteratively. Eventually, when movement of any RN cannot further increase the network lifetime, we declare that the M RNs are placed at optimal locations and the algorithm terminates.

The proposed SPINDS algorithm consists of two phases: 1) SP and 2) INDS and works as follows. Initially, we put all RNs at the BS B .⁷ At the beginning of each iteration, we first obtain the best flow routing under these RNs' locations. Then, we estimate the lifetime of each node⁸ (include AFNs and RNs) and order the nodes in increasing order of node lifetime. We identify the node with the smallest node lifetime through this process and denote it as node i . Note that node i can be either an AFN or an RN.

Suppose that node i is an AFN. In this case, we consider node i as the center point relative to other nodes and denote it as O . We then make a list of all RNs in the order of increasing distance to point O . Then, we pair node i with an RN, say r_j , that is farthest away from point O in the list of RNs. This is the SP step in SPINDS.

⁷If we have better position that we know *a priori*, we can start by putting RNs at these locations. This will help speed up the algorithm's running time.

⁸This is done based on current incoming and outgoing flows at each node.

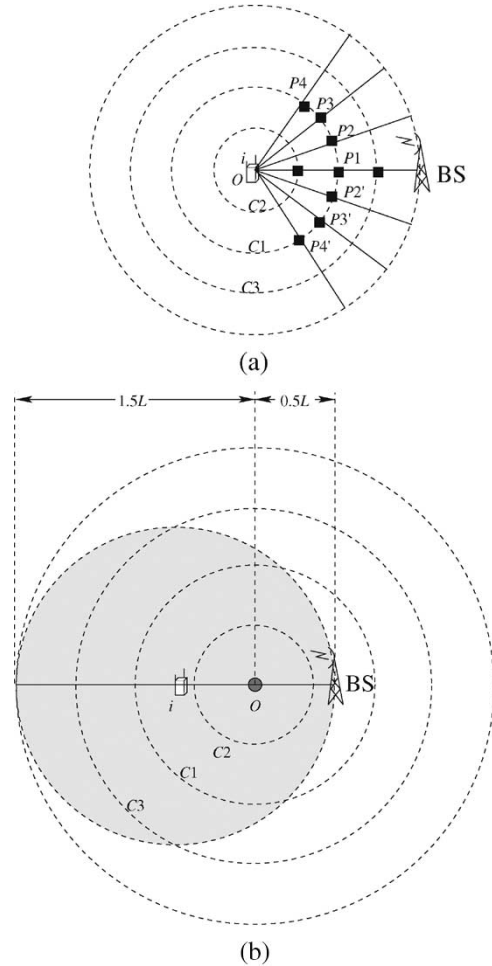


Fig. 4. INDS step in the SPINDS algorithm. (a) Node i is an AFN. (b) Node i is an RN.

Once an RN r_j is paired with node O , we attempt to move this RN to a better location within the disc region where the disc is centered at point O and has a radius (say L) equal to the distance between point O and BS [see Fig. 4(a)]. Note that it is sufficient to search this disc area (with radius L) for RN since AFN i would reach the BS with a shorter distance if an RN is outside this disc. It is also necessary to search the entire area of the disc (instead of only the segment between O and B). This is because that we are not interested in the increase of any individual node's lifetime, but rather the lifetime of the entire network. An increase of network lifetime will need the collaboration of rearranging the flow routing topology among all N AFNs and M potential RNs, which means that any point on the disc could be a potential candidate to place this RN and make an improvement in network lifetime.

Since AFN i has finite energy, the closer the RN r_j moves to AFN i , the longer the lifetime of AFN i can be prolonged. The closest position to AFN i , in the extreme case, is point O itself. Therefore, we first try to put the RN r_j to coincide with point O , which corresponds to the situation that energy will be provisioned on AFN i directly. With this placement, if the network lifetime is increased, we are done. Otherwise, the possible distance from r_j to i is in $(R_l, R_u) = (0, L)$. We search a circle C_1 having radius $R = (R_l + R_u)/2 = L/2$. In

particular, we start from point P_1 [see Fig. 4(a)] and move along the circle C_1 with equal phase angle θ . That is, we try points (P_2, P'_2) , (P_3, P'_3) , (P_4, P'_4) , and so forth on the circle C_1 over 360° . If the network lifetime increases when the RN is placed at any of these new points on the circle C_1 , we update R_u by R and move to circle C_2 ; otherwise (no network lifetime improvement), we update R_l by R and move to circle C_3 . Again, the radius of the new circle is $R = (R_l + R_u)/2$. Then, we repeat the search process for the points on the new circle as we have done for circle C_1 . Clearly, the radius of each circle involved in the search process resembles a binary search. Eventually, the search terminates if $R_u - R_l$ is less than a threshold δ_L . This is the so-called INDS step.

The case when node i is an RN is similar to that for the case when i is an AFN, except that the center of the disc O is now defined as the midpoint between RN i and the BS [see Fig. 4(b)]. The reason why we choose this midpoint as the disc center is as follows. Since node i is an RN, its energy is therefore also adjustable. Thus, the lifetime of RN i can be prolonged by adding more energy. A good starting point to place an RN (from the viewpoint of RN i) would be the midpoint between i and B , which we choose as the disc center O . Since point O is the center point, we organize the RNs (excluding RN i) in a list in increasing distance toward point O and designate the RN having the largest distance toward point O as RN r_j . We pair (i, r_j) together and start INDS. Although the search region is still a disc centered at i with radius L [shaded area in Fig. 4(b)], where L is the distance between i and B , the center of the search circle is now O . Hence, the largest circle that covers the shaded disc has a radius of $1.5L$. However, when we search points on circles C_1 , C_2 , C_3 , and so forth for a better RN placement point, it is only necessary to search the portion of the circle that lies within the shaded disc area. The radius of each circle involved in the search process also resembles a binary search.

At the end of each iteration, we will either have moved the RN to a new location and obtained an increased network lifetime (as well as an EP strategy) or have no improvement and thus this RN r_j will not be moved. In the former case, we move on to the next iteration (start with a new set of pairing). In the latter case, we drop RN r_j from the current pairing and choose the next RN in the list of RNs, which is the RN that is second farthest away from point O . Should all the RNs have been considered for pairing, we move on to the next node on the list with the second smallest node lifetime and perform the same pairing and search process. The algorithm terminates when the network lifetime cannot be further improved after pairing all the nodes with all the RNs during an iteration. The pseudo-code for the SPINDS algorithm is given in Fig. 5.

Substituting MILP With LP: Note that for each placement trial for RN r_j , we need to solve an EP(AFN+RN) problem, which is an MILP. Due to repetitive routine calls by SPINDS, the computational burden would be prohibitively high. Fortunately, Lemma 1 shows that if the heuristic algorithm is designed wisely, then solving a simpler EP(RN) (which is an LP)

<p>Notation for the SPINDS algorithm.</p> <ol style="list-style-type: none"> 1. NS: A stack of nodes, including N AFNs and M RNs. 2. In NS, nodes closer to the top have smaller lifetimes. 3. O: Center of search disc. 4. If current pairing node i is an AFN, O is i; 5. else O is the mid-point between i and B; 6. RS_i: A stack of RNs that is dynamically formed according to the current pairing node. 7. In RS, nodes closer to the top are farther away from O. 8. $[R_l, R_u]$: The search space for radius. 9. δ_L: The threshold for radius change. 10. θ: The phase (or angle) increment during a search on a circle.
<p>SP in the SPINDS algorithm.</p> <ol style="list-style-type: none"> 1. Initialization: 2. Put all M RNs at the base-station B; 3. increase=1; 4. Solve an EP(RN) problem to obtain the optimal T, energy allocation, and flow routing; 5. while (increase=1) { 6. Reset increase=0; 7. for ($i = 1$; $i \leq N + M$; $i++$) 8. Estimate node i's lifetime; 9. Sort nodes in non-increasing order of its estimated node lifetime and arrange the sorted list with a stack NS; 10. while ($(NS \neq \text{NULL})$ and (increase==0)) { 11. $i = \text{pop}(NS)$; 12. Sort all RNs in non-decreasing order of distance to O and arrange the sorted list with a stack RS_i; 13. (if i is an RN, i will not be included in RS_i) 14. while ($RS_i \neq \text{NULL}$) { 15. $j = \text{pop}(RS_i)$; 16. increase=INDS(i, j); 17. if (increase=1) 18. break; } } 19. break; } }
<p>INDS in the SPINDS algorithm.</p> <ol style="list-style-type: none"> 1. int INDS(int i, int j) { 2. relocate=0; 3. if (i is an AFN) 4. $R_u = d_{iB}$; 5. else // i.e., i is an RN 6. $R_u = d_{iB} * 1.5$; 7. Try to put RN r_j at point O; 8. If network lifetime increases { 9. Relocate r_j to point O; 10. return 1; } 11. $R_l=0$; 12. while ($R_u - R_l > \delta_L$) { 13. $R = (R_l + R_u)/2$; 14. Try to put RN r_j on the circle centered at O and with radius R for every θ degree; 15. //When i is an RN, if $d_{ij} > d_{iB}$, no need to try this point 16. If network lifetime increases { 17. Relocate RN r_j to the new position; 18. relocate=1; 19. $R_u=R$; } 20. else $R_l=R$; } 21. return relocate; } 22. }

Fig. 5. Implementation of SPINDS algorithm.

would yield the same result. Thus, we will use the simpler EP(RN) computation for each RN r_j placement decision during each iteration of SPINDS. It is not hard to show that the SPINDS algorithm terminates in polynomial time.

IV. NUMERICAL INVESTIGATION

In this section, we present results from our numerical investigation. First, we use a simple network example (with known optimal solution) and show that SPINDS can indeed offer good results after going through its algorithmic iterations. Then, we demonstrate the performance of SPINDS for general network configurations and compare it to some other approaches. We will also provide important insights on the EP-RNP problem.

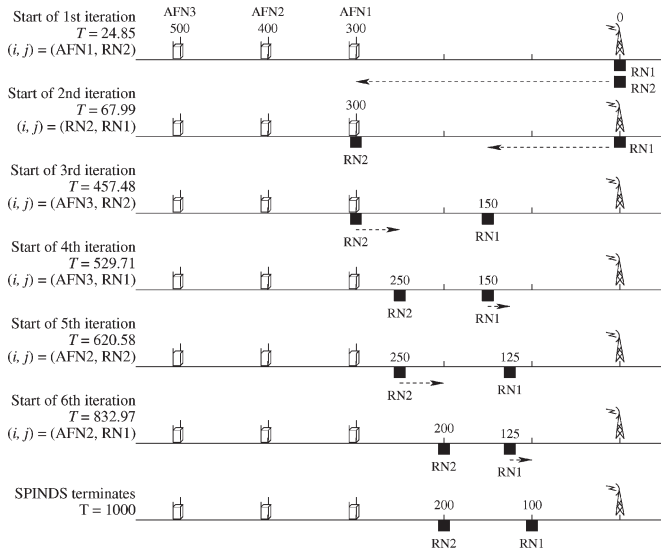


Fig. 6. Example illustrating the iterations of SPINDS. The unit for lifetime T is in 1000×10^4 s.

A. A Simple Example

To show how SPINDS can indeed offer good solutions to the EP-RNP problem, we use the following simple one-dimensional test network to demonstrate the iterative behavior of SPINDS. Suppose that we have three AFNs that are located along a line away from the BS (see Fig. 6). Assume that the position for the BS is location 0. AFN_1 , AFN_2 , and AFN_3 are located 300, 400, and 500 m away from the BS, respectively. The initial energy and local data generation rate for each respective AFN are $e_1 = 6.4$ kJ, $g_1 = 1$ kb/s; $e_2 = 4.1$ kJ, $g_2 = 1$ kb/s; $e_3 = 1.8$ kJ, $g_3 = 1$ kb/s. Now, suppose that the total amount of provisioning energy E is 13.8 kJ and that we can deploy up to two RNs ($M = 2$). By using back-of-the-envelope calculations, it can be easily found that the optimal places for these RNs should be 100 and 200 m away from the BS, respectively, and on the same line with the other AFNs. Also, each RN should be provisioned with an amount of energy of 6.9 kJ. The maximum achievable network lifetime is $T = 1000 \times 10^4$ s.

Now, we apply the SPINDS algorithm to the same problem. Fig. 6 shows the iterative steps in placing RNs for each iteration. As the example shows, SPINDS terminates to the optimal result after six iterations.

B. Performance of SPINDS

We will use the 10-AFN, 20-AFN, and 50-AFN network topologies for our numerical investigation. Without loss of generality, in all network topologies, we assume that the BS is at the origin point (0, 0) (in meters). Tables II–IV give each AFN’s location (x_i, y_i) (in meters), local data generating rate g_i [in kilobit per second (kb/s)], and initial energy e_i [in kilojoule (kJ)] for each topology, respectively, all of which are generated randomly. The amount of available provisioning energy for the 10-AFN, 20-AFN, and 50-AFN networks are 1000 kJ, 1400 kJ, and 500 kJ, respectively, which are also set randomly.

TABLE II
LOCATIONS, DATA GENERATING RATE, AND INITIAL ENERGY OF EACH AFN FOR A 10-AFN NETWORK

AFN	(x_i, y_i) (m)	g_i (kb/s)	e_i (kJ)
1	(45,450)	2	18.63
2	(-50,90)	10	49.68
3	(-455,475)	7	83.60
4	(460,-85)	9	62.20
5	(0,-480)	2	15.76
6	(-500,-485)	5	87.58
7	(-485,40)	1	45.51
8	(100,-490)	8	52.27
9	(470,-495)	1	16.38
10	(95,-40)	3	95.69

TABLE III
LOCATIONS, DATA GENERATING RATE, AND INITIAL ENERGY OF EACH AFN FOR A 20-AFN NETWORK

AFN	(x_i, y_i) (m)	g_i (kb/s)	e_i (kJ)
1	(100,50)	1	10.75
2	(-400,-400)	3	13.01
3	(-100,400)	1	14.88
4	(10,410)	8	63.61
5	(-410,420)	1	99.89
6	(-90,100)	1	14.42
7	(400,-500)	10	51.57
8	(410,-300)	2	19.47
9	(90,460)	7	15.95
10	(500,500)	9	29.60
11	(50,-100)	6	25.52
12	(-80,0)	3	74.86
13	(200,460)	5	37.86
14	(300,-490)	5	59.42
15	(-420,-480)	5	86.85
16	(490,-200)	9	69.02
17	(190,-480)	9	64.33
18	(-500,200)	6	90.87
19	(-490,-90)	4	6.230
20	(10,460)	4	4.608

First, we will examine the impact of the parameters (θ and δ_L) for SPINDS. For the parameter δ_T , we set it to 100 s. This value is much smaller than the final network lifetime and is appropriate for all practical situations. The settings for θ and δ_L reflect the tradeoff between computational complexity and precision of search space. Fig. 7(a) shows, when $\delta_L = 50$ m is fixed, the network lifetime obtained by SPINDS under different θ for the 10-AFN network. As expected, the smaller the θ , the better the network lifetime performance. An important observation in Fig. 7(a) is that, when $\theta \leq 30^\circ$, the improvement in network lifetime is negligible. Therefore, we choose $\theta = 30^\circ$ in all of our numerical results. Fig. 7(b) shows, when $\theta = 30^\circ$ is fixed, the network lifetime obtained by SPINDS under different δ_L for the same 10-AFN network. Again, as expected, the smaller δ_L is, the longer the network lifetime can become. Similar to the case for θ , when $\delta_L \leq 50$ m, the improvement in network lifetime is negligible. As a result, we choose $\delta_L = 50$ m for all numerical results.

To demonstrate the performance of SPINDS, we compare it with two other approaches to the EP-RNP problem. The first approach is the greedy incremental (GI) algorithm and is based on the following simple idea. Although it is not computationally feasible to perform an exhaustive search for placing M RNs simultaneously, it is possible to choose an optimal

TABLE IV
LOCATIONS, DATA GENERATING RATE, AND INITIAL ENERGY FOR EACH AFN FOR THE 50-AFN NETWORK

AFN	(x_i, y_i) (m)	g_i (kb/s)	e_i (kJ)	AFN	(x_i, y_i) (m)	g_i (kb/s)	e_i (kJ)	AFN	(x_i, y_i) (m)	g_i (kb/s)	e_i (kJ)
1	(-300,320)	2	83.2	18	(-390,330)	8	67.2	35	(60,500)	7	80.6
2	(340,-420)	8	69.9	19	(290,430)	2	61.0	36	(-40,350)	6	100.0
3	(-120,290)	4	91.2	20	(-160,-380)	9	56.9	37	(-80,130)	6	92.8
4	(250,470)	5	100.0	21	(-10,-350)	4	60.8	38	(-200,490)	10	56.9
5	(-270,-220)	5	96.5	22	(-450,-250)	4	93.9	39	(420,-440)	6	89.9
6	(-10,430)	7	58.3	23	(-340,-410)	9	69.7	40	(-70,-120)	3	62.5
7	(0,300)	7	97.6	24	(-190,-390)	3	80.7	41	(-70,-100)	5	52.3
8	(-470,-340)	9	84.4	25	(-320,0)	9	60.7	42	(420,330)	8	95.2
9	(500,-90)	3	85.6	26	(-400,-180)	4	93.9	43	(-60,-400)	9	64.7
10	(190,500)	8	99.9	27	(60,-140)	10	84.5	44	(-340,300)	3	92.4
11	(-320,-440)	3	61.7	28	(-300,300)	3	54.6	45	(-490,180)	2	79.0
12	(-200,-390)	9	73.5	29	(-200,-260)	6	96.8	46	(240,140)	9	76.6
13	(430,220)	2	63.0	30	(-160,-420)	7	69.5	47	(110,-250)	2	76.2
14	(-80,-350)	5	73.3	31	(320,-360)	7	79.7	48	(390,490)	2	93.2
15	(330,460)	7	86.3	32	(-380,-310)	9	83.5	49	(-60,400)	4	89.1
16	(60,30)	7	100.0	33	(-370,-110)	5	80.4	50	(410,-400)	6	69.0
17	(180,120)	3	92.3	34	(-220,190)	8	64.2	—			

position to place one RN at a time. The best location for placing one node can be found by exhaustively searching all tiny grids that are drawn within the feasible region. Once the location for this RN is fixed, we can place the next RN following the same process. Under this approach, the RNs are placed one by one until all M potential RNs are placed. We choose the grid size to be 10×10 m for the 1000×1000 network dimension, which corresponds to 10^4 grids.

Another approach that we use in comparison is to provision the available energy E only on the existing N AFNs without deploying additional RNs. In this approach, since M provisioning points can only be chosen from the existing N AFNs, we must have $M \leq N$.⁹ Note that the constraint $M \leq N$ does not apply to SPINDS or GI, where M can exceed N .

Fig. 8(a) shows, given the total available energy $E = 1000$ kJ, the maximum network lifetime obtained under different EP approaches for the 10-AFN network. For the SPINDS and GI approaches, we also used both LP and MILP in the solution process. There are several important observations from this figure. First, we note that for the SPINDS algorithm, the numerical results using the MILP and LP match closely with each other. Recall that in Lemma 1, if we can choose the M -node placement points wisely, without losing critical points in the search space, then the solution obtained by solving the simple EP(RN) problem will yield the same result as that obtained by solving the EP(AFN+RN) problem. Therefore, we see that the SPINDS algorithm indeed explores search space wisely, thereby justifying the use of the LP (and thus polynomial) instead of the MILP in our algorithmic design.

Second, we examine the greedy incremental (GI) approach using both the MILP and the LP techniques. Clearly, the LP technique for the GI approach is considerably worse than the MILP approach. This is mainly due to the fact that, under greedy algorithm, the locations for RNs deployed during earlier iterations cannot be changed in future iterations. As a result,

although the location for each individual RN is best chosen during each incremental placement, the locations for the M RNs, when they are considered jointly, are poorly chosen. Consequently, Lemma 1 would not be applicable here and we conclude that the GI approach cannot offer good solutions for EP-RNP with LP techniques.

Third, under the EP(AFN) approach where there is no RN and additional energy can only be added on the existing AFNs, the network lifetime performance is very poor compared to SPINDS. Even when M increases, the increase in network lifetime is still very small. This phenomena conclusively demonstrates that there indeed exists deficiencies in this network topology and EP on existing AFNs alone cannot mitigate this problem and bring much improvement in network lifetime performance. In this case, RN placement is the only viable approach to fundamentally mitigate network geometric deficiency and prolong network lifetime.

Finally, we find that under the same total amount of available energy, the number of RNs can have a significant impact on the overall network lifetime performance. For example, in Fig. 8(a), under SPINDS, the network lifetime can increase 65-fold as the number of EP points (M) increases from 1 to 15 under the same total provisioning energy of 1000 kJ.

In Fig. 8(b), we conduct a similar investigation for the 20-AFN network. The numerical results for this 20-AFN network reaffirm all of our observations for the ten-AFN network.

To explore the performance limits of RNP, in Fig. 8(c), we plot the network lifetime performance for a $N = 50$ node network under three different approaches.¹⁰ For the $N = 50$ node network, the geometric deficiency problem is less of an issue compared to the $N = 10$ and $N = 20$ networks discussed earlier. As a result, we suspect that the improvement of SPINDS over other approaches may not be very significant. In Fig. 8(c), we find that SPINDS is still noticeably better than

¹⁰Note that the complexities of the SPINDS with MILP approach and the GI with MILP approach are both too high for numerical computation. Therefore, their results are not shown in Fig. 8(c)

⁹This is the so-called EP(AFN) problem in Fig. 3.

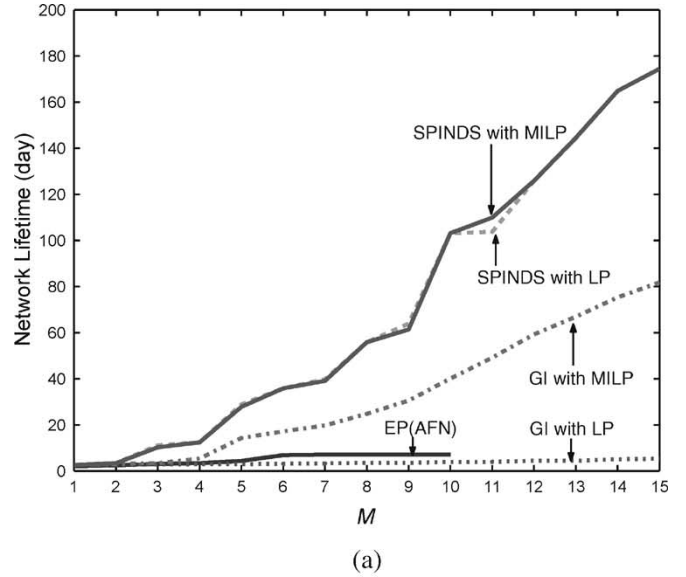
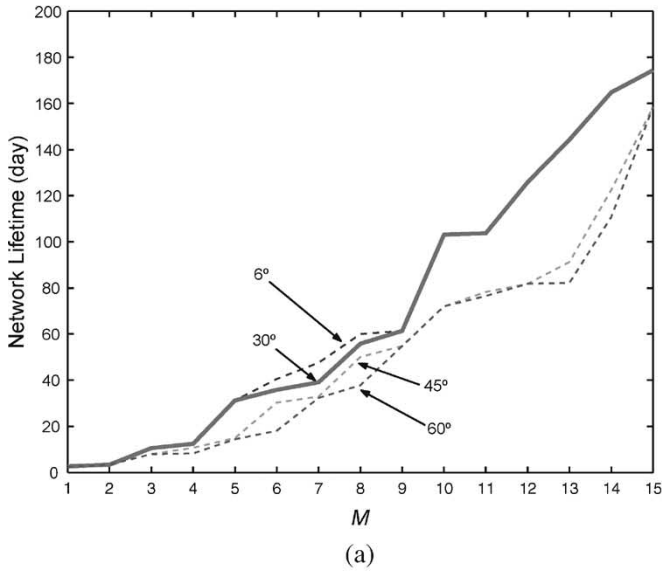


Fig. 7. Network lifetime results of SPINDS from various parameters. (a) Parameter θ . (b) Parameters δ_L .

GI when the total number of nodes in the network ($M + N$) is less than 70 (or $M \leq 20$). For $M > 20$, or the total number of nodes in the network exceeds 70 ($N + M > 70$), the difference in network lifetime performance between SPINDS and GI diminishes. Specifically, both SPINDS and GI tend to reach a saturation point as the number of RNs increases. The interpretation for these phenomena is that when the network density becomes sufficiently high, all of its geometric deficiency will be effectively mitigated (even under GI approach). As a result, once above a density threshold, the network lifetime will reach a saturation point over which RNP can no longer further increase this lifetime limit. For the network under consideration, the network lifetime limit is approximately 89 days. Even under this scenario, there is still the advantage of using SPINDS over GI. This is because SPINDS tends to approach this limit much faster than GI. In particular, with only $M = 15$ RNs, SPINDS can almost reach this limit, while under GI, it will take at least $M = 30$ RNs.

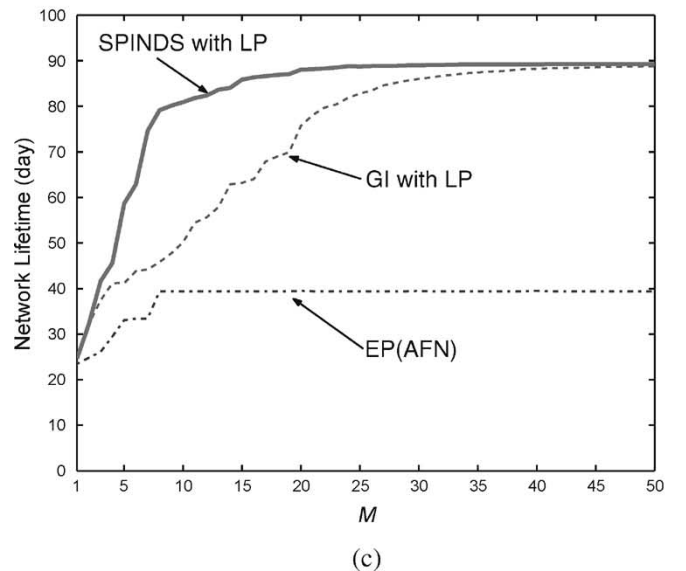
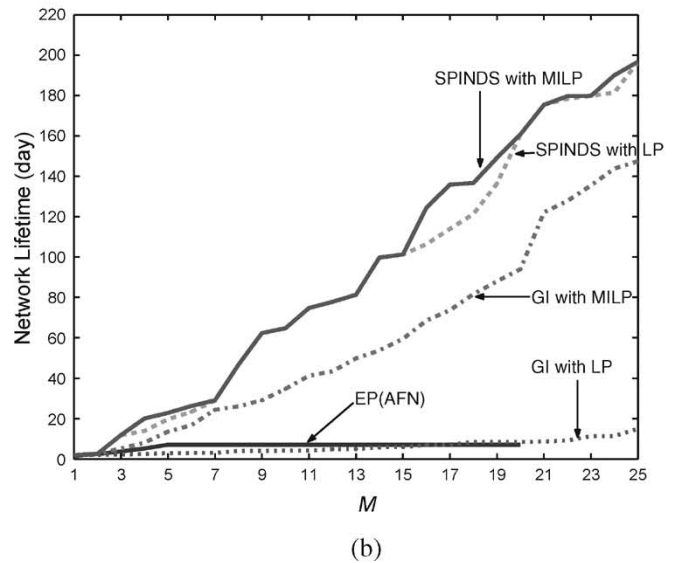


Fig. 8. Network lifetime results from various EP approaches and computational techniques. (a) A 10-AFN network. (b) A 20-AFN network. (c) A 50-AFN network.

V. RELATED WORK

At the time of this work, there is no known prior work that directly addresses the EP and RNP problems for wireless sensor networks. Nevertheless, we will briefly review related research on power control and network lifetime, which motivated us to pursue this line of research.

On the network layer, most work on the power control problem can be classified into one of two categories. The first class comprises strategies of finding an optimal transmitter power to control the connectivity properties of the network (see, e.g., [12], [16], [18], and [25]). The second class of approaches could be called power “aware” routing. Most schemes use the shortest path algorithm with a power-based metric rather than a hop-count-based metric (see, e.g., [14], [15], and [22]). However, power-based shortest path routing does not ensure good performance in energy-constrained applications (e.g., network lifetime). Using power-based shortest paths may result in premature depletion of energy at certain nodes, which is not optimal in network lifetime.

Maximizing network lifetime based on power control has been explored in several recent works. In particular, Chang and Tassiilas [4] formulate the network lifetime problem as an LP problem, which is similar to what we have done to calculate network lifetimes for the EP(RN) problem. But none of these prior efforts have addressed the EP and RNP problem, which is the focus on this paper.

VI. CONCLUSION

In this paper, we investigated the important problem of energy provisioning (EP) for wireless sensor networks. We considered a two-tier wireless sensor network and studied the joint problem of EP and relay node placement (EP–RNP) for the upper tier aggregation and forwarding nodes (AFNs) to increase network lifetime. Since the EP–RNP problem formulation is NP-hard, we developed an efficient polynomial-time heuristic algorithm, smart pairing and Intelligent Disc Search (SPINDS), that solves the EP–RNP problem. SPINDS is an iterative algorithm that attempts to increase the network lifetime by iteratively moving an RN to a better location. The polynomial running time property of SPINDS was achieved by transforming the original mixed-integer nonlinear programming (MINLP) problem into an iterative LP problem. Through numerical results, we showed that the proposed SPINDS is highly competitive in solving the EP–RNP problem when compared with some other approaches. We also offered some important insights on network geometric properties, RN placement, and energy provisioning.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their thorough review and constructive comments that helped improve several aspects of this paper. The authors also wish to thank Prof. N. Sahinidis of the University of Illinois, Urbana-Champaign, for his assistance in using the BARON software.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *Comput. Networks (Elsevier)*, vol. 38, no. 4, pp. 393–422, 2002.
- [2] BARON Global Optimization Software. [Online]. Available: <http://archimedes.scs.uiuc.edu/baron/baron.html>
- [3] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 2nd ed. New York: Wiley, 1993.
- [4] J.-H. Chang and L. Tassiilas, “Energy conserving routing in wireless ad-hoc networks,” in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 26–30, 2000, pp. 22–31.
- [5] J. Chou, D. Petrovis, and K. Ramchandran, “A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks,” in *Proc. IEEE INFOCOM*, San Francisco, CA, Mar. 30–Apr. 3, 2003, pp. 1054–1062.
- [6] M. A. Duran and I. E. Grossmann, “An outer-approximation algorithm for a class of mixed-integer nonlinear programs,” *Math. Program.*, vol. 36, no. 3, pp. 307–339, 1986.
- [7] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.
- [8] A. M. Geoffrion, “Generalized benders decomposition,” *Optim. Theory Appl.*, vol. 10, no. 4, pp. 237–260, Oct. 1978.
- [9] O. K. Gupta and A. Ravindran, “Branch and bound experiments in convex nonlinear integer programming,” *Manage. Sci.*, vol. 31, no. 12, pp. 1533–1546, 1985.
- [10] W. Heinzelman, “Application-specific protocol architectures for wireless networks,” Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Mass. Inst. Technol., Cambridge, MA, Jun. 2000.
- [11] Y. T. Hou, Y. Shi, J. Pan, and S. F. Midkiff, “Maximizing lifetime of wireless sensor networks through optimal single-session flow routing,” *IEEE Trans. Mobile Comput.*, submitted for publication.
- [12] L. Hu, “Topology control for multihop packet radio networks,” *IEEE Trans. Commun.*, vol. 41, no. 10, pp. 1474–1481, Oct. 1993.
- [13] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” in *Proc. 16th Annu. ACM Symp. Theory Computing*, Washington, DC, 1984, pp. 302–311.
- [14] Q. Li, J. Aslam, and D. Rus, “Online power-aware routing in wireless ad-hoc networks,” in *Proc. ACM Mobile Computing and Networking (Mobicom)*, Rome, Italy, Jul. 16–21, 2001, pp. 97–107.
- [15] A. Michail and A. Ephremides, “Energy efficient routing for connection-oriented traffic in ad-hoc wireless networks,” in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Radio Communication (PIMRC)*, London, U.K., Sep. 2000, pp. 762–766.
- [16] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, “Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW Protocol,” in *Proc. European Wireless Conf.—Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, Florence, Italy, Feb. 2002, pp. 156–162.
- [17] S. S. Pradhan, J. Kusuma, and K. Ramchandran, “Distributed compression in a dense sensor network,” *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 51–60, Mar. 2002.
- [18] R. Ramanathan and R. Rosales-Hain, “Topology control of multihop wireless networks using transmit power adjustment,” in *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 26–30, 2000, pp. 404–413.
- [19] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Upper Saddle River, NJ: Prentice-Hall, 1996.
- [20] V. Rodoplu and T. H. Meng, “Minimum energy mobile wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1333–1344, Aug. 1999.
- [21] A. Sayeed, D. Estrin, G. Pottie, and K. Ramchandran, *Guest Editors, IEEE Journal on Selected Areas in Communications—Special Issue on Self-organizing Distributed Collaborative Sensor Networks*, to be published.
- [22] S. Singh, M. Woo, and C. S. Raghavendra, “Power-aware routing in mobile ad hoc networks,” in *Proc. ACM Mobile Computing and Networking (Mobicom)*, Dallas, TX, Oct. 1998, pp. 181–190.
- [23] D. Slepian and J. K. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 471–480, Jul. 1973.
- [24] M. Tawarmalani, “Mixed integer nonlinear programs: theory, algorithms and applications,” Ph.D. dissertation, Dept. Mech. Ind. Eng., Univ. Illinois, Urbana-Champaign, Aug. 2001.
- [25] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, “Distributed topology control for power efficient operation in multihop wireless ad hoc networks,” in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 22–26, 2001, pp. 1388–1397.

- [26] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 1, pp. 1–10, Jan. 1976.



Y. Thomas Hou (S'91–M'98–SM'04) received the B.E. degree in electrical engineering from the City College of New York, New York, in 1991, the M.S. degree in electrical engineering from Columbia University, New York, in 1993, and the Ph.D. degree in electrical engineering from the Polytechnic University, Brooklyn, NY, in 1998.

From 1997 to 2002, he was a Research Scientist and Project Leader at Fujitsu Laboratories of America, IP Networking Research Department, Sunnyvale, CA (Silicon Valley). Since Fall 2002, he has

been an Assistant Professor at the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, Virginia. His research interests are in algorithmic design and optimization for network systems. His current research focuses on wireless sensor networks and multimedia over wireless ad hoc networks. In recent years, he has worked on scalable architectures, protocols, and implementations for differentiated services Internet; service overlay networking; multimedia streaming over the Internet; and network bandwidth allocation policies and distributed flow control algorithms. He has published extensively in the above areas.

Dr. Hou was the corecipient of the 2002 IEEE International Conference on Network Protocols (ICNP) Best Paper Award and the 2001 IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (CSVT) Best Paper Award. He is a Member of the Association for Computing Machinery (ACM).



Yi Shi (S'02) received the B.S. degree in computer science from the University of Science and Technology of China, Hefei, China, in 1998, the M.S. degree in computer science from the Institute of Software, Chinese Academy of Science, Beijing, China, in 2001, the second M.S. degree in computer science from the Virginia Polytechnic Institute and State University (Virginia Tech), Blacksburg, VA, in 2003, and is currently working toward the Ph.D. degree in electrical and computer engineering at Virginia Tech.

His current research focuses on algorithms and optimization for wireless sensor networks and wireless ad hoc networks. His work has appeared in some highly selective international conferences such as the Association for Computing Machinery International Conference on Mobile Computing and Networking (ACM Mobicom), the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), and the IEEE Conference on Computer Communications (INFOCOM).

Mr. Shi was the recipient of the Meritorious Award in International Mathematical Contest in Modeling in 1997 and 1998, respectively.



Hanif D. Sherali is a W. Thomas Rice Endowed Chaired Professor of Engineering at the Industrial and Systems Engineering Department at the Virginia Polytechnic Institute and State University, Blacksburg, VA. His area of research interest is in discrete and continuous optimization with applications to location, transportation, and engineering design problems. He has published about 200 papers in Operations Research journals, has coauthored four books in this area, and serves on the editorial board of eight journals.

Mr. Sherali is a Member of the National Academy of Engineering.



Scott F. Midkiff (S'82–M'85–SM'92) received the M.S. degree in electrical engineering from Stanford University, Stanford, CA, and the B.S.E. and Ph.D. degrees in electrical engineering from Duke University, Durham, NC.

He joined the Bradley Department of Electrical and Computer Engineering at the Virginia Polytechnic Institute and State University, Blacksburg, VA, in 1986 and is currently a Professor. He previously worked at Bell Laboratories and held a visiting position at Carnegie Mellon University. His research

interests include system issues in wireless and ad hoc networks, network services for pervasive computing, and performance modeling of mobile ad hoc networks.