Constrained Relay Node Placement in Wireless Sensor Networks: Formulation and Approximations

Satyajayant Misra, *Member, IEEE*, Seung Don Hong, Guoliang (Larry) Xue, *Senior Member, IEEE*, and Jian Tang, *Member, IEEE*

***Abstract*—One approach to prolong the lifetime of a wireless sensor network (WSN) is to deploy some *relay nodes* to communicate with the sensor nodes, other relay nodes, and the base stations. The relay node placement problem for wireless sensor networks is concerned with placing a minimum number of relay nodes into a wireless sensor network to meet certain connectivity or survivability requirements. Previous studies have concentrated on the *unconstrained* version of the problem in the sense that relay nodes can be placed anywhere. In practice, there may be some physical constraints on the placement of relay nodes. To address this issue, we study *constrained* versions of the relay node placement problem, where relay nodes can only be placed at a set of candidate locations. In the connected relay node placement problem, we want to place a minimum number of relay nodes to ensure that each sensor node is connected with a base station through a bidirectional path. In the survivable relay node placement problem, we want to place a minimum number of relay nodes to ensure that each sensor node is connected with two base stations (or the only base station in case there is only one base station) through two node-disjoint bidirectional paths. For each of the two problems, we discuss its computational complexity and present a framework of polynomial time $\mathcal{O}(1)$-approximation algorithms with small approximation ratios. Extensive numerical results show that our approximation algorithms can produce solutions very close to optimal solutions.**

***Index Terms*—Approximation algorithms, connectivity and survivability, relay node placement, wireless sensor networks (WSNs).**

I. INTRODUCTION

A WIRELESS sensor network (WSN) consists of many low-cost and low-power *sensor nodes* (SNs) [1]. Sensing and short-range communication to transmit sensed information to the base stations are two of the most important functions of a SN in a WSN. There has been extensive research on energy-aware routing [4], [14], [18], [33], improvement in lifetime [12], [24], [32], and survivability [22] in WSNs. Since energy consumption is proportional to d^κ for transmitting over distance d, where κ is a constant in the interval $[2, 4]$, long-distance transmission in WSNs is costly. To prolong network lifetime while meeting certain network specifications, researchers have proposed to deploy in a WSN a small number of *relay nodes* (RNs) whose main function is to communicate with the SNs and other RNs [2], [5], [11], [12], [15], [20], [21], [32]. These are studied under the general theme of *relay node placement*. Recently, this problem has received a lot of attention from the networking community, with papers addressing this problem published in MobiCom [24], MobiHoc [2], [29], and Infocom [10], [15], [34].

Relay node placement problems can be classified into either *single-tiered* or *two-tiered* based on the routing structures [11], [12], [21], [24] and into either *connected* or *survivable* based on the connectivity requirements [2], [11], [15], [34]. In single-tiered relay node placement, a SN also forwards packets received from other nodes. In two-tiered relay node placement, a SN forwards its sensed information to a RN or a base station (BS), but does not forward packets received from other nodes. In connected relay node placement, we place a small number of RNs to ensure that each sensor node is connected with a base station through a bidirectional path. In survivable relay node placement, we place a small number of RNs to ensure that each sensor node is connected with two base stations (or the only base station, in case there is only one base station) through two node-disjoint bidirectional paths.

We first briefly review prior works on single-tiered relay node placement, where both relay nodes and sensor nodes participate in the forwarding of received packets. We will use R and r to denote the communication ranges of RNs and SNs, respectively. We will also use $k = 1$ to denote connectivity requirement and use $k \geq 2$ to denote survivability requirement. In 1999, Lin and Xue [19] studied the problem with $R = r$ and $k = 1$, proved its NP-hardness, and presented a minimum spanning tree (MST) based 5-approximation algorithm. They also designed a *steinerization scheme*, which has been used by almost all later works [2], [3], [5], [10], [11], [15], [20], [21], [29], [34]. Chen *et al.* [3] proved that the Lin–Xue algorithm is a 4-approximation algorithm and presented a 3-approximation algorithm. Cheng *et al.* [5] presented a faster 3-approximation algorithm and a randomized 2.5-approximation algorithm. Bredin *et al.* [2] extended the relay node placement problem to the case of $R = r$ and $k \geq 2$ and presented polynomial time $\mathcal{O}(1)$-approximation algorithms for any fixed k. Kashyap *et al.* [15] presented a 10-approximation algorithm for the case of $R = r$ and $k = 2$. All of the

Manuscript received July 16, 2008; revised March 20, 2009; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Liu. First published November 24, 2009; current version published April 16, 2010. This work was supported in part by NSF Grants 0901451 and 0830739 and ARO Grant W911NF-04-1-0385. The information reported here does not reflect the position or the policy of the federal government. This is an extended and enhanced version of the paper [23] that appeared in IEEE INFOCOM 2008.

S. Misra is with the Department of Computer Science, New Mexico State University, Las Cruces, NM 88003 USA (e-mail: misra@cs.nmsu.edu).

S. D. Hong and G. Xue are with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: seung.hong@asu.edu; xue@asu.edu).

J. Tang is with the Department of Computer Science, Montana State University, Bozeman, MT 59717 USA (e-mail: tang@cs.montana.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2009.2033273

above works assume that the transmission range of the RNs is the same as that of the SNs. Lloyd and Xue [21] studied the problem with $R \geq r$ and $k = 1$, proved its NP-hardness, and presented a 7-approximation algorithm. Zhang *et al.* [34] presented a 14-approximation algorithm for $R \geq r$ and $k = 2$.

We next give a brief review of prior works on two-tiered relay node placement, where only the RNs participate in packet forwarding. Motivated by the works [9] and [24] on clustered WSNs, Hao *et al.* in [11] formulated the two-tiered relay node placement problems where each SN has to be within distance r of at least k RNs, and the RNs (all having communication range $R \geq r$) form a k-connected network, for $k = 1, 2$. Tang *et al.* in [30] presented 4.5-approximation algorithms for $k = 1$ and 2, under the assumption that $R \geq 4r$ and that the SNs are uniformly distributed. In [20], under the assumption that $R = r$ and with no restriction on the distribution of the SNs, Liu *et al.* presented a $(6 + \epsilon)$-approximation algorithm for $k = 1$ and a $(24+\epsilon)$-approximation algorithm for $k = 2$, where $\epsilon > 0$ is any given constant. In [21], Lloyd and Xue studied the problem for $k = 1$ with the condition $R = r$ relaxed to $R \geq r$ and presented a $(5+\epsilon)$-approximation algorithm. Srinivas *et al.* [29] presented better approximation algorithms under the assumption $R \geq 2r$. Zhang *et al.* [34] studied both single-tiered and two-tiered relay node placement problems that ensure 2-connectivity, which involves sensor nodes, relay nodes, and base stations. They presented $\mathcal{O}(1)$-approximation algorithms for both problems.

All of the above works study *unconstrained* relay node placement in the sense that the RNs can be placed *anywhere*. For example, in the works [2], [21], and [34], the relay nodes are stacked on top of other relay nodes or sensor nodes. In practice, however, there may be some physical constraints on the placement of the RNs. For example, there may be a lower bound on the distance between two network nodes to reduce interference. Also, there may be some forbidden regions where relay nodes cannot be placed. However, the relay node placement problem subject to forbidden regions and lower bound on internode distance is intrinsically harder than its unconstrained counterpart. As a first step toward solving this challenging problem, we study *constrained* relay node placement problems where the RNs can only be placed at a set of candidate locations. Our formulation can be viewed as an approximation to the aforementioned relay node placement problem subject to the constraints of forbidden regions and lower bound on internode distance in the following sense. Instead of allowing the relay nodes to be placed *anywhere* outside of the forbidden regions and satisfying the internode distance bound, we further restrict the placement of the relay nodes to certain candidate locations that are outside of the forbidden regions. The use of candidate locations simultaneously approximates the constraint enforced by the forbidden regions and the constraint enforced by the internode distance bound. We are using a discrete optimization problem to approximate a nonconvex continuous optimization problem.

In this paper, we study single-tiered constrained relay node placement problems under both the connectivity requirement and the survivability requirement. We formulate the problems, discuss their complexities, and present polynomial time $\mathcal{O}(1)$-approximation algorithms. To our best knowledge, we are the first to present $\mathcal{O}(1)$-approximation algorithms for these problems.

In Section II, we present basic notations and prove a few fundamental lemmas that will be used in later sections. In Section III, we study the *connected* relay node placement problem. In Section IV, we study the *survivable* relay node placement problem. In Section V, we present linear programming-based schemes for computing lower bounds on the optimal solutions of the relay node placement problems. We present numerical results in Section VI and conclude the paper in Section VII.

II. Basic Notations and Fundamental Lemmas

We consider a *hybrid wireless sensor network* (HWSN) consisting of SNs, RNs, and BSs. We assume that all SNs have communication range $r > 0$ and that all RNs have communication range $R \geq r$, where r and R are given constants. We also assume that the BSs are powerful enough so that their communication range is much greater than R, and that any two BSs can communicate directly with each other. We note that, in practice, two BSs may communicate indirectly via other means, such as satellites or the Internet. Since the objective of this paper is to place the minimum number of RNs to meet connectivity or survivability requirements, our assumption simplifies notations without losing any generality. We use $d(x, y)$ to denote the Euclidean distance between two points x and y in the plane. We will also use u to denote the location of a node u, if no confusion arises.

Following the above discussions, two nodes u and v can communicate directly with each other if and only if $d(u, v)$ is less than or equal to the smaller of the communication ranges of the two nodes. In other words, a SN u can communicate directly with another node v (which could be a SN, a RN, or a BS) if and only if $d(u, v) \leq r$. A RN u can communicate directly with another node v (which could be a RN or a BS) if and only if $d(u, v) \leq R$. Similarly, any pair of BSs can communicate directly with each other. Following these rules, the SNs, the RNs, and the BSs, together with the values of r and R, collectively induce a *hybrid communication graph* (HCG) formally defined in the following.

Definition 2.1: Let $\mathcal{B}$ be a set of BSs, $\mathcal{X}$ be a set of SNs, $\mathcal{Y}$ be a set of RNs, and $R \geq r > 0$ be the respective communication ranges of RNs and SNs. The *hybrid communication graph* $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y})$ induced by the 5-tuple $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y})$ is an undirected graph with node set $V = \mathcal{B} \cup \mathcal{X} \cup \mathcal{Y}$ and edge set E defined as follows. For any two BSs $b_i, b_j \in \mathcal{B}$, E contains the undirected edge $(b_i, b_j) = (b_j, b_i)$. For a RN $y \in \mathcal{Y}$ and a node $z \in \mathcal{B} \cup \mathcal{Y}$, which could be either a RN or a BS, E contains the undirected edge $(y, z) = (z, y)$ if and only if $d(y, z) \leq R$. For a SN $x \in \mathcal{X}$ and a node $z \in \mathcal{B} \cup \mathcal{X} \cup \mathcal{Y}$, which is either a SN, a RN, or a BS, E contains the undirected edge $(x, z) = (z, x)$ if and only if $d(x, z) \leq r$. □

We illustrate the concept of HCG using the example shown in Fig. 1(a). In this example, the set of SNs is $\mathcal{X} = \{x_1, x_2\}$, the set of RNs is $\mathcal{Y} = \{y_1, y_2\}$, and the set of BSs is $\mathcal{B} = \{b_1, b_2\}$. Therefore, the HCG has six vertices. There is an edge (x_1, y_1) in the HCG because $d(x_1, y_1) \leq r$. Similarly, the HCG also

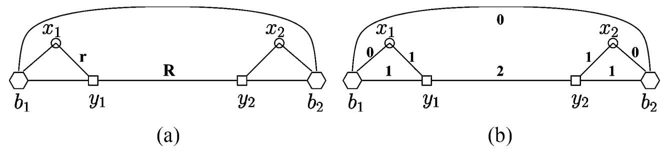


Fig. 1. (a) Illustration of HCG, showing HCG$(r, R, \{b_1, b_2\}, \{x_1, x_2\}, \{y_1, y_2\})$, where $d(x_1, b_1) = d(x_1, y_1) = d(x_2, b_2) = d(x_2, y_2) = r$, $d(y_1, y_2) = R$. (b) Edge weights in HCG, where an edge incident with no relay node has a weight of 0, an edge incident with exactly one relay node has a weight of 1, and an edge incident with two relay nodes has a weight of 2.

contains the edges (x_1, b_1), (x_2, b_2), and (x_2, y_2). There is an edge (y_1, y_2) in the HCG connecting RNs y_1 and y_2 because $d(y_1, y_2) \leq R$. Similarly, the HCG also contains the edges (y_1, b_1) and (y_2, b_2). There is an edge (b_1, b_2) in the HCG connecting BSs b_1 and b_2 because we assume any pair of BSs are directly connected.

The hybrid communication graph defines all possible bidirectional communications between pairs of nodes. For the design and analysis of our schemes, we will need to define two more concepts related to an HCG, i.e., the *edge weights* and the *relay size* of an HCG. These are formally defined in the following. We use the following standard graph theoretic notations: for a graph G, $V(G)$ denotes the node set of G and $E(G)$ denotes the edge set of G.

Definition 2.2: Let $G = \mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y})$ be a hybrid communication graph. For each edge $e = (u, v)$ in the HCG, we define its *weight* (denoted by $w(e)$) as

$$w(e) = |\{u, v\} \cap \mathcal{Y}|. \tag{2.1}$$

Let H be a subgraph of G. The weight of H [denoted by $w(H)$] is defined as

$$w(H) = \sum_{e \in E(H)} w(e). \tag{2.2}$$

The relay size of H [denoted by $s(H)$] is defined as

$$s(H) = |V(H) \cap \mathcal{Y}|. \tag{2.3}$$

In other words, the relay size of H is the number of relay nodes in H. □

Fig. 1(b) illustrates the edge weights of the HCG shown in Fig. 1(a). Simply put, the weight $w(u, v)$ of an edge in the HCG is the number of relay nodes the edge is incident with. Recall that our goal is to use the minimum number of relay nodes to ensure that the sensor nodes and the base stations are connected or biconnected. This assignment of edge weight in the HCG ensures that a constant approximation to a minimum weight subgraph of HCG connecting all of the sensor nodes and the base stations corresponds to a constant approximation to an optimal solution of the connected relay node placement problem, and that a constant approximation to a minimum weight 2-connected subgraph of HCG connecting all of the sensor nodes and the base stations corresponds to a constant approximation to an optimal solution of the survivable relay node placement problem. More precisely, our definition of the weight and relay size of a subgraph of an HCG leads to an important relationship between the weight and the relay size of a certain class of subgraphs of an HCG, which is stated in the following lemma.

Lemma 2.1: Let H be a subgraph of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y})$ such that every RN in H has degree at least 2 (within H). Then, $w(H) \geq 2 \cdot s(H)$. □

Proof: We prove this lemma by *shifting the edge weight to its end nodes*. Initially, every node in H has its weight initialized to 0. We loop over all edges of H to move the edge weights to their end nodes in the following way.

Let (u, v) be an edge of H that is incident with two RNs. According to our definition, the weight of this edge is 2. In this case, we divide the edge weight into two equal pieces, add a weight of 1 to node u, and add a weight of 1 to node v. Let (u, v) be an edge of H, where u is a RN and v is not. According to our definition, the weight of this edge is 1. In this case, we add a weight of 1 to node u, add a weight of 0 to node v. Let (u, v) be an edge of H where neither u nor v is a RN. According to our definition, the weight of this edge is 0. In this case, we add a weight of 0 to node u and add a weight of 0 to node v.

After all edges are looped over, we have shifted the edge weights of H to the RNs in H. Note that a relay node u is getting a weight of 1 from every edge of H that is incident with u, resulting in a weight equal to the degree of u. Since every RN in H is incident with at least two edges in H, it receives a weight of at least 2. Therefore, $w(H) \geq 2 \cdot s(H)$. ■

We use Fig. 1(b) to illustrate Lemma 2.1 and its proof. Assume that H is the HCG in Fig. 1(a). We have $w(H) = 1 + 1 + 2 + 1 + 1 = 6$ and $s(H) = 2$. Clearly, we have $w(H) = 6 \geq 2 \cdot s(H) = 4$. Following the *weight shifting scheme* used in the proof, RN y_1 receives a weight of 1 from edge (x_1, y_1), a weight of 1 from edge (b_1, y_1), and a weight of 1 from edge (y_2, y_1), resulting in a total weight of 3 (≥ 2). Similarly, RN y_2 receives a weight of 1 from edge (x_2, y_2), a weight of 1 from edge (b_2, y_2), and a weight of 1 from edge (y_1, y_2), resulting in a total weight of 3 (≥ 2). Therefore, each RN receives a weight that is equal to its degree, as stated in the proof of Lemma 2.1.

We will also need to use the result stated in Lemma 2.3, which is based on Lemma 2.2 in the following.

Lemma 2.2: Let $G(V, E)$ be an undirected biconnected graph where $|V| \geq 3$ and each edge $e \in E$ has a unit length $l(e) = 1$. Let $H(V, E')$ be a minimum length biconnected subgraph of G. Then, $|E'| \leq 2|V| - 3$. □

Proof: Since $G(V, E)$ is biconnected, we can find an *ear decomposition* of G [31]. Let H be defined by all the ears in an ear decomposition of G. Then, H is a biconnected subgraph of G spanning all vertices in V. We need to prove that H contains no more than $2|V| - 3$ edges.

By definition, the first ear is a cycle spanning n_1 (≥ 3) vertices and contains n_1 edges. Each additional ear spans n_i (≥ 1) new vertices using $n_i + 1$ edges. Therefore, the total number of edges in H is at most $2|V| - 3$. ■

Lemma 2.3: Let $G(V, E)$ be an undirected connected graph where $|V| \geq 3$ and each edge $e \in E$ has a unit length $l(e) = 1$. Let $H(V, E')$ be a minimum length connected subgraph of G such that two vertices u and v are in the same biconnected component of H if and only if they are in the same biconnected component of G. Then, $|E'| \leq 2|V| - 1$. □

Proof: Let $H_1, \ldots, H_k$ be the biconnected components of H, where H_i has $n_i \geq 3$ vertices, $i = 1, 2, \ldots, k$. Note that two biconnected components H_i and H_j may share one common node, but never two. Assume that the union of $H_1, \ldots, H_k$ has p connected components $(1 \leq p \leq k)$. Let $V \backslash (H_1 \cup \cdots \cup H_k) = \{v_1, v_2, \ldots, v_q\}$, where $q = |V| - \sum_{i=1}^{k} n_i + (k-p)$. Then, H can be obtained by connecting the p connected components of the union of $H_1, \ldots, H_k$ and q vertices $\{v_1, v_2, \ldots, v_q\}$ using exactly $p+q-1$ edges in G. Therefore, the number of edges in H is

$$|E'| = \sum_{i=1}^{k} |E(H_i)| + (p+q-1) \tag{2.4}$$

$$\leq \sum_{i=1}^{k} (2n_i - 3) + (p+q-1) \tag{2.5}$$

$$= 2|V| - 1 - (k+p+q) \leq 2|V| - 1 \tag{2.6}$$

where the second equality follows from $p + q - k = |V| - \sum_{i=1}^{k} n_i$. This proves the lemma. ■

Definition 2.3: Let $G = \mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y})$ be a hybrid communication graph. Let H be a subgraph of G. Let u be a relay node in H. The *sensor degree of* u in H, denoted by $\delta_s(u, H)$, is the number of SNs that are neighbors of u in H. The *base station degree of* u in H, denoted by $\delta_b(u, H)$, is the number of BSs that are neighbors of u in H. The *maximum sensor degree of* H is defined as $\Delta_s(H) = \max\{\delta_s(u, H) | u \in V(H) \cap \mathcal{Y}\}$. The *maximum base station degree of* H is defined as $\Delta_b(H) = \max\{\delta_b(u, H) | u \in V(H) \cap \mathcal{Y}\}$. The *maximum nonrelay degree of* H is defined as $\Delta(H) = \max\{\delta_b(u, H) + \delta_s(u, H) | u \in V(H) \cap \mathcal{Y}\}$. □

It is clear that $\Delta(H) \leq \Delta_s(H) + \Delta_b(H)$. For graph theoretic terms not defined in this paper, we refer readers to the standard textbook [31]. We will use (u, v) to denote the undirected edge in a graph. Therefore, (u, v) and (v, u) denote the same edge. We will use the terms *nodes* and *vertices* interchangeably, as well as *links* and *edges*. For concepts in algorithms and computing theory, such as *NP-hard*, we refer readers to the standard textbooks [6], [8].

A polynomial time β-approximation algorithm for a minimization problem is an algorithm $\mathcal{A}$ that, for any instance of the problem, computes a solution that is at most β times the optimal solution of the instance in time bounded by a polynomial in the input size of the instance [6]. In this case, we also say that $\mathcal{A}$ has an approximation ratio of β.

III. Relay Node Placement to Ensure Connectivity

Given a set of SNs, a set of BSs, and a set of *candidate locations where* RNs *can be placed*, we are interested in *placing the minimum number of* RNs so that the sensor nodes and the base stations are in the same connected component of the hybrid communication graph induced by the SNs, the RNs, and the BSs.

Relay node placement in wireless sensor networks has been studied by many researchers [2], [3], [5], [10], [11], [15], [19]–[21], [29], [34]. The objective here is to shift the load of long-distance transmissions from the SNs to the RNs, therefore achieving better energy efficiency and extending network lifetime. Most of the previous studies have concentrated on the case where the RNs can be placed *anywhere*. In practice, however, there are certain restrictions on the locations of the RNs with respect to the SNs, the BSs, and other RNs. This motivated us to study the *constrained relay node placement problem*. In this section, we study the problem of placing the minimum number of RNs to ensure network connectivity.

A. Problem Definitions and Discussions

Definition 3.1: Let $R \geq r > 0$ be the respective communication ranges for RNs and SNs. Let $\mathcal{B}$ be a set of BSs, $\mathcal{X}$ be a set of SNs, and $\mathcal{Z}$ be a set of candidate locations where RNs can be placed. A set of RNs $\mathcal{Y} \subseteq \mathcal{Z}$ is said to be a *feasible connected relay node placement* (denoted by $\mathcal{F}$-RNPc) for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ if the graph $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y})$ is connected. The *size* of the corresponding $\mathcal{F}$-RNPc is $|\mathcal{Y}|$. A $\mathcal{F}$-RNPc is said to be a *minimum connected relay node placement* for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ (denoted by $\mathcal{M}$-RNPc) if it has the minimum size among all $\mathcal{F}$-RNPc for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. □

Definition 3.2: Let $R \geq r > 0$ be the respective communication ranges for RNs and SNs. Let $\mathcal{B}$ be a set of BSs, $\mathcal{X}$ be a set of SNs, and $\mathcal{Z}$ be a set of candidate locations where RNs can be placed. The *connected relay node placement problem* for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, denoted by **RNPc-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, seeks a $\mathcal{M}$-RNPc for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. □

We also study a special case of **RNPc-P** where $\mathcal{B} = \emptyset$. Many existing works correspond to this special case [3], [5], [19], [21]. For this special case, our algorithm has a faster running time and a better approximation ratio. In this case, one of the relay nodes deployed can work as a base station to collect data.

The authors of [16] studied the **Critical-Grid Coverage Problem** (**CGCP**). **CGCP** is concerned with a grid of equilateral triangles of size ℓ. Some of the grid points are *critical grids*, which need to be covered by sensor nodes with sensing range R_s and communication range R_t. The goal of **CGCP** is to deploy a minimum number of sensor nodes on the grid points so that: 1) each critical-grid point is within distance R_s of at least one sensor node; and 2) the sensor nodes form a connected network under the unit disk communication model, where two sensor nodes can communicate iff they are within distance R_t. It is proved in [16] that if $\ell \leq R_s < 2\ell/\sqrt{3}$ and $2\ell \leq R_t$, then **CGCP** is NP-hard using a reduction from planar 3SAT.

We demonstrate in the following that the special case of **RNPc-P** with $\mathcal{B} = \emptyset$ contains **CGCP** as a special case. For any given instance $\mathcal{I}$ of **CGCP**, we construct an instance $\mathcal{I}'$ of **RNPc-P** with $\mathcal{B} = \emptyset$ in the following way. Let $\mathcal{X}$ (the set of sensor nodes in $\mathcal{I}'$) be the set of critical grids in $\mathcal{I}$. Let $\mathcal{Z}$ (the candidate locations for relay nodes in $\mathcal{I}'$) be the rest of the grid points contained in the convex hull of the critical grids. Let $r = R_s$, $R = R_t$, and $\mathcal{B} = \emptyset$. Then, we have an instance $\mathcal{I}'$ for **RNPc-P**. Moreover, an optimal solution to $\mathcal{I}'$ corresponds to an optimal solution to $\mathcal{I}$, and an α-approximation to $\mathcal{I}'$ corresponds to an α-approximation to $\mathcal{I}$. Since **CGCP** is NP-hard, **RNPc-P** is also NP-hard. Therefore, we seek efficient algorithms that have provably good performance guarantees.

We present a general *framework* of efficient approximation algorithms for **RNPc-P**, based on efficient approximation algorithms for the graph Steiner tree problem (**STP**) [13]. In particular, we show that by using the best-known approximation algorithm for **STP** [28], our framework becomes a $\mathbf{5.5}$-approximation algorithm for **RNPc-P** when $\mathcal{B} = \emptyset$, and a $\mathbf{6.2}$-approxima-

TABLE I
CLOSELY RELATED RESULTS ON CONNECTED RELAY NODE PLACEMENT

source	connectivity	R vs r	$\mathcal{B} \neq \emptyset$	constraints	approx ratio
[19]	1	$R = r$			5
[3]	1	$R = r$			3
[5]	1	$R = r$			3
[21]	1	$R \geq r$			7
this	1	$R \geq r$		$\surd$	**5.5**
this	1	$R \geq r$	$\surd$	$\surd$	**6.2**

tion algorithm for the general **RNPc-P**. To the best of our knowledge, we are the first to present $\mathcal{O}(1)$-approximation algorithms for these constrained relay node placement problems. The *unconstrained version* of **RNPc-P** when $\mathcal{B} = \emptyset$ is the *single-tiered relay node placement problem* ($\mathbf{1tRNP}$) studied by Lloyd and Xue [21], where there is no restriction on the locations of the relay nodes. Considering that the best-known approximation algorithm [21] for $\mathbf{1tRNP}$ (the unconstrained problem) has an approximation ratio of $\mathbf{7}$, our $\mathbf{5.5}$-approximation algorithm for the constrained problem is amazingly good. Table I lists the most closely related results on this topic.

B. A Framework of Efficient Approximation Algorithms

In this section, we present a framework of polynomial time approximation algorithms for **RNPc-P**. For the general case, we prove that the number of RNs used by our algorithm is no more than 4β times the number of RNs required by an optimal solution, where β is the approximation ratio of the approximation algorithm $\mathcal{A}$ for **STP**. For the special case where $\mathcal{B} = \emptyset$, we prove that the number of RNs used by our algorithm is no more than 3.5β times the number of RNs required by an optimal solution. Our approximation algorithm for **RNPc-P** is presented as Algorithm 1.

Algorithm 1 Approximation for **RNPc-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$

Input: $R \geq r > 0$, set of BSs $\mathcal{B}$, set of SNs $\mathcal{X}$, set of candidate locations of RNs $\mathcal{Z}$, and an approximation algorithm $\mathcal{A}$ for the **STP**.

Output: An $\mathcal{F}$-RNPc for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ given by $\mathcal{Y}_{\mathcal{A}} \subseteq \mathcal{Z}$.

1: Construct $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$.

2: **if** the nodes in $\mathcal{B} \cup \mathcal{X}$ are not in a single connected component of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ **then**

3: The **RNPc-P** instance does not have a feasible solution. Stop.

4: **end if**

5: Assign edge weights to the edges in $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ as in Definition 2.2.

6: Apply algorithm $\mathcal{A}$ to compute a low weight tree subgraph $\mathcal{T}_{\mathcal{A}}$ of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ which connects all nodes in $\mathcal{B} \cup \mathcal{X}$.

7: Output $\mathcal{Y}_{\mathcal{A}} = \mathcal{Z} \cap V(\mathcal{T}_{\mathcal{A}})$.

Algorithm 1 takes as input an instance of **RNPc-P** and an approximation algorithm $\mathcal{A}$ for the graph **STP**. Simply put, an instance of **STP** is given by an undirected graph $G(V, E, w)$ and a subset $X \subseteq V$ of *target nodes*, where V and the E are the set of nodes and edges, respectively, and $w(e) \geq 0$ is the *weight* of edge $e \in E$. The goal is to compute a minimum weight tree subgraph $\mathcal{T}$ of G such that $\mathcal{T}$ connects all target nodes (maybe some additional nodes in $V \setminus X$, which are called Steiner nodes) [13]. This problem is NP-hard, but admits many polynomial time approximation algorithms with small constant approximation ratios. The simplest approximation algorithm for **STP** is the MST-based 2-approximation algorithm of Kou *et al.* [17]. The algorithm first constructs an edge-weighted complete graph $\mathcal{C}(X)$ on the set X of target nodes, where the weight of an edge (u, v) in $\mathcal{C}(X)$ is the weight of the minimum weight u–v path in G. We then compute an MST of $\mathcal{C}(X)$. Since each edge (u, v) in $\mathcal{C}(X)$ corresponds to a u–v path in G, the MST of the complete graph corresponds to a connected subgraph G' of G. Note that G' is not necessarily a tree. We can then compute a tree subgraph of G' and repeatedly delete nontarget leaf nodes in the tree. The resulting tree is a Steiner tree whose cost is no more than twice that of the optimal Steiner tree. Other more sophisticated approximation algorithms are also known. For example, with a longer running time (still a polynomial time algorithm), the algorithm of [28] has an approximation ratio of $1 + \frac{\ln 3}{2} \leq 1.55$.

The major steps of Algorithm 1 are as follows. First, we construct the hybrid communication graph $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, as if we were placing an RN at every candidate location in $\mathcal{Z}$. This is accomplished in Line 1 of the algorithm. It should be noted that the given instance of the problem has a feasible solution if and only if all the BSs and SNs are in the same connected component of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. Recall that we assume that all base stations are connected. We can compute all of the connected components of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ in linear time using depth first search [6]. This is accomplished in Lines 2–4 of the algorithm. Next, we assign nonnegative integer weights to the edges of the HCG as in Definition 2.2, i.e., the weight of an edge is the number of relay nodes with which it is incident. This is accomplished in Line 5 of the algorithm. Then, we apply algorithm $\mathcal{A}$ to compute a low-weight tree subgraph $\mathcal{T}_{\mathcal{A}}$ of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, spanning all nodes in $\mathcal{B} \cup \mathcal{X}$. This is accomplished in Line 6 of the algorithm. Finally, in Line 7, we identify the locations to place the RNs.

We use the example shown in Fig. 2 to illustrate Algorithm 1. Fig. 2(a) shows six SNs (illustrated using small circles), two BSs (illustrated using small hexagons), and 18 candidate locations for RNs (illustrated using small squares). These 26 nodes are sitting on unit grid points. Assuming $r = 1$ and $R = 2$, the edges of the corresponding HCG are also shown, where the 0-weight edges (edges with weight 0) are shown in dashed lines, the 1-weight edges are shown in dash-dot lines, and the 2-weight edges are shown in solid lines. Fig. 2(b) shows the edge-weighted complete graph on $\mathcal{B} \cup \mathcal{X}$, where the weight of an edge in the complete graph is the length of the shortest path connecting the two end nodes in the HCG. A MST of the complete graph is shown in thick edges. Fig. 2(c) shows the relay node placement corresponding to the MST, which uses six RNs,

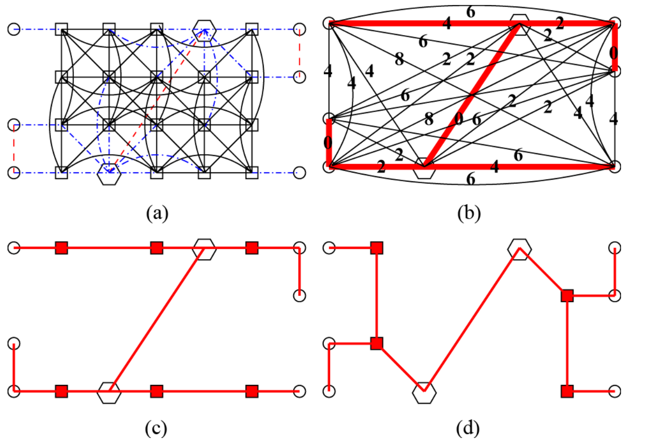


Fig. 2. (a) The HCG of the instance for two BSs (hexagons), six SNs (circles), and 18 candidate locations for RNs (squares). (b) The edge-weighted complete graph on $\mathcal{B} \cup \mathcal{X}$, where the edge weight is the shortest path length in the HCG, and a MST (thick edges). (c) The corresponding MST-based $\mathcal{F}$-RNPc, which uses six RNs. (d) The optimal solution, which uses four RNs.

shown as filled squares. Fig. 2(d) shows the optimal relay node placement, which uses four RNs.

Theorem 3.1: Algorithm 1 has a worst-case running time bounded by $O(|\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|^2 + T(\mathcal{A}))$, where $T(\mathcal{A})$ is the time complexity of the approximation algorithm $\mathcal{A}$ used for approximating the **STP** problem. Furthermore, we have the following:

- **RNPc-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ has a feasible solution if and only if HCG $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ has a connected component that contains all nodes in $\mathcal{B} \cup \mathcal{X}$.
- When **RNPc-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ has a feasible solution, Algorithm 1 guarantees computing a feasible solution that uses no more than $\frac{\beta}{2}(\Delta(\mathcal{T}_{\text{opt}}) + 2)$ times the number of RNs required in an optimal solution $\mathcal{Y}_{\text{opt}}$, where β is the approximation ratio of $\mathcal{A}$, and $\mathcal{T}_{\text{opt}}$ is a minimum spanning tree of HCG$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{\text{opt}})$. □

Proof: Line 1 constructs the HCG, which requires $O(|\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|^2)$ time. Lines 2–4 can be accomplished using depth first search, which also requires $O(|\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|^2)$ time. Line 5 also requires $O(|\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|^2)$ time. Line 6 requires $O(T(\mathcal{A}))$ time. This proves the time complexity of the algorithm.

If not all the nodes in $\mathcal{B} \cup \mathcal{X}$ are in the same connected component of HCG$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, there must be two nodes $u, v \in \mathcal{B} \cup \mathcal{X}$ that are not connected in HCG$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. Since all base stations are connected with each other in the HCG, this means that there is at least one sensor node that is not connected to a base station, implying that the given instance does not have a feasible solution. On the other hand, if all the nodes in $\mathcal{B} \cup \mathcal{X}$ are in the same connected component of HCG$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, any tree subgraph of HCG$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ which spans all the nodes in $\mathcal{B} \cup \mathcal{X}$ corresponds to an $\mathcal{F}$-RNPc of the given instance.

Let $\mathcal{T}_{\min}$ be a minimum weight tree subgraph of HCG$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ which connects all nodes in $\mathcal{B} \cup \mathcal{X}$. Since $\mathcal{T}_{\text{opt}}$ is a tree subgraph of HCG$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ which connects all nodes in $\mathcal{B} \cup \mathcal{X}$, we have

$$w(\mathcal{T}_{\min}) \leq w(\mathcal{T}_{\text{opt}}). \tag{3.1}$$

Since $\mathcal{A}$ is a β-approximation algorithm, we have

$$w(\mathcal{T}_{\mathcal{A}}) \leq \beta \cdot w(\mathcal{T}_{\min}) \leq \beta \cdot w(\mathcal{T}_{\text{opt}}). \tag{3.2}$$

We can write $w(\mathcal{T}_{\text{opt}})$ as $w(\mathcal{T}_{\text{opt}}) = w_1(\mathcal{T}_{\text{opt}}) + w_2(\mathcal{T}_{\text{opt}})$, where $w_1(\mathcal{T}_{\text{opt}})$ is the sum of the 1-weight edges in $\mathcal{T}_{\text{opt}}$ and $w_2(\mathcal{T}_{\text{opt}})$ is the sum of the 2-weight edges in $\mathcal{T}_{\text{opt}}$. Since $\Delta(\mathcal{T}_{\text{opt}}) \geq \delta_s(u, \mathcal{T}_{\text{opt}}) + \delta_b(u, \mathcal{T}_{\text{opt}})$ for each RN u in $\mathcal{Y}_{\text{opt}}$

$$w_1(\mathcal{T}_{\text{opt}}) \leq \Delta(\mathcal{T}_{\text{opt}}) \cdot |\mathcal{Y}_{\text{opt}}|. \tag{3.3}$$

Since $\mathcal{T}_{\text{opt}}$ is a tree, it has at most $|\mathcal{Y}_{\text{opt}}| - 1$ 2-weight edges. Therefore

$$w_2(\mathcal{T}_{\text{opt}}) \leq 2 \cdot (|\mathcal{Y}_{\text{opt}}| - 1). \tag{3.4}$$

Therefore

$$w(\mathcal{T}_{\text{opt}}) \leq (2 + \Delta(\mathcal{T}_{\text{opt}})) \cdot |\mathcal{Y}_{\text{opt}}| - 2. \tag{3.5}$$

Combining Lemma 2.1 and inequalities (3.2) and (3.5), we have

$$\begin{aligned} |\mathcal{Y}_{\mathcal{A}}| &\leq \frac{1}{2} w(\mathcal{T}_{\mathcal{A}}) \leq \frac{\beta}{2} w(\mathcal{T}_{\text{opt}}) \\ &\leq \frac{\beta}{2}(2 + \Delta(\mathcal{T}_{\text{opt}}))|\mathcal{Y}_{\text{opt}}|. \end{aligned} \tag{3.6}$$

This proves the theorem. ■

There are several choices for the approximation algorithm $\mathcal{A}$. For example, if we use the algorithm of [17], the corresponding approximation ratio is $\beta = 2$. If we use the algorithm of [28], the corresponding approximation ratio is $\beta = 1 + \frac{\ln 3}{2} \leq 1.55$. Next, we will prove a bound on $\Delta(\mathcal{T}_{\text{opt}})$.

Lemma 3.1: Let $\mathcal{T}_{\text{opt}}$ be a MST of HCG$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{\text{opt}})$, where $\mathcal{Y}_{\text{opt}}$ is an optimal solution to **RNPc-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. Then, $\Delta_s(\mathcal{T}_{\text{opt}}) \leq 5$ and $\Delta_b(\mathcal{T}_{\text{opt}}) \leq 1$. □

Proof: We prove this by contradiction. Assume that in $\mathcal{T}_{\text{opt}}$, a RN u is connected to six SNs v_1, v_2, v_3, v_4, v_5, v_6. Without loss of generality, assume that $\angle v_1 u v_2 \leq 60^o$. Since $d(u, v_1) \leq r$ and $d(u, v_2) \leq r$, we have $d(v_1, v_2) \leq r$. Since $\mathcal{T}_{\text{opt}}$ is a tree, it does not contain edge (v_1, v_2), as otherwise there would be a cycle (u, v_1, v_2, u). Replacing edge (u, v_1) in $\mathcal{T}_{\text{opt}}$ with edge (v_1, v_2), we obtain another tree $\mathcal{T}_1$ spanning the nodes $\mathcal{B} \cup \mathcal{X} \cup \mathcal{Y}$. Since $w(u, v_1) = 1$ and $w(v_1, v_2) = 0$, we have $w(\mathcal{T}_1) < w(\mathcal{T}_{\text{opt}})$, contradicting the assumption that $\mathcal{T}_{\text{opt}}$ is a minimum spanning tree. Therefore, an RN u cannot be connected to more than five SNs in $\mathcal{T}_{\text{opt}}$.

Now, assume that a relay node u is connected to two BSs b_1 and b_2 in $\mathcal{T}_{\text{opt}}$. Since $\mathcal{T}_{\text{opt}}$ is a tree, it does not contain the edge (b_1, b_2). We can replace edge (u, b_1) in $\mathcal{T}_{\text{opt}}$ with edge (b_1, b_2) to obtain another lower weight tree $\mathcal{T}_2$ spanning the nodes $\mathcal{B} \cup \mathcal{X} \cup \mathcal{Y}$. This contradiction proves that no RN in $\mathcal{T}_{\text{opt}}$ can be connected to more than one BSs. ■

Corollary 3.1: The general **RNPc-P** has a polynomial time **6.2**-approximation algorithm. The special case of **RNPc-P** with $\mathcal{B} = \emptyset$ has a polynomial time **5.5**-approximation algorithm. □

Proof: According to Robins and Zelikovsky [28], there is a polynomial time approximation scheme for the **STP** whose approximation ratio can be made arbitrarily close to $1 + \frac{\ln 3}{2} <$

1.55. The claims of this corollary follow from Theorem 3.1 with $\beta = 1.55$ and the $\Delta(\mathcal{T}_{\text{opt}})$ bound derived in Lemma 3.1. ■

Corollary 3.2: The general **RNPc-P** has an **8**-approximation algorithm with a running time of $O(|\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|^2 \log |\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|)$. The special case of **RNPc-P** (with $\mathcal{B} = \emptyset$) has a **7**-approximation algorithm with a running time of $O(|\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|^2 \log |\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|)$. □

Proof: If we take $\mathcal{A}$ in Algorithm 1 as the MST-based 2-approximation algorithm for **STP** [17], the running time of Algorithm 1 is $O(|\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|^2 \log |\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|)$. The corresponding approximation ratios of Algorithm 1 follows from Theorem 3.1 and Lemma 3.1. ■

IV. Relay Node Placement to Ensure Survivability

In Section III, we have studied the relay node placement problem under the connectivity requirement, i.e., there is a path connecting every pair of nodes $u, v \in \mathcal{B} \cup \mathcal{X}$. In this section, we consider a relay node placement problem that meets *both the connectivity requirement and the survivability requirement*. In particular, we need to ensure that between each pair of nodes $u, v \in \mathcal{B} \cup \mathcal{X}$, there exists a pair of node-disjoint paths connecting u and v.

Survivable relay node placement (also known as fault-tolerant relay node placement) in wireless sensor networks has been studied by many researchers [2], [10], [11], [15], [20], [29], [34]. The objective here is to ensure that the network remains connected in the presence of up to $K \geq 1$ node failures. For a network to tolerate up to K node failures, it has to be $K+1$-connected. The works [11], [15], [20], [29], and [34] study relay node placement that ensures 2-connectivity, while the works [2] and [10] study relay node placement that ensures higher order connectivity. All these works can be viewed as *unconstrained survivable relay node placement* in the sense that relay nodes can be placed anywhere. Our current work can be viewed as *constrained survivable relay node placement* in the sense that relay nodes can only be placed at some prespecified candidate locations.

A. Problem Definitions and Discussions

Given a set of SNs, a set of BSs, as well as the candidate locations where RNs can be placed, we are interested in *placing the minimum number of relay nodes* so that the hybrid communication graph induced by the SNs, the RNs, and the BSs is biconnected.

Definition 4.1 : Let $R \geq r > 0$ be the respective communication ranges for RNs and SNs. Let $\mathcal{B}$ be a set of BSs, $\mathcal{X}$ be a set of SNs, and $\mathcal{Z}$ be a set of candidate locations where RNs can be placed. A set of RNs $\mathcal{Y} \subseteq \mathcal{Z}$ is said to be a *feasible survivable relay node placement* (denoted by $\mathcal{F}$-RNPs) for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ if the graph $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y})$ is biconnected. The *size* of the corresponding $\mathcal{F}$-RNPs is $|\mathcal{Y}|$. A $\mathcal{F}$-RNPs is said to be a *minimum survivable relay node placement* for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ (denoted by $\mathcal{M}$-RNPs) if it has the minimum size among all $\mathcal{F}$-RNPs for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. □

Definition 4.2: Let $R \geq r > 0$ be the respective communication ranges for RNs and SNs. Let $\mathcal{B}$ be a set of BSs, $\mathcal{X}$ be a set of SNs, and $\mathcal{Z}$ be a set of candidate locations where RNs can be placed. The *survivable relay node placement problem* for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, denoted by **RNPs-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, seeks a $\mathcal{M}$-RNPs for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. □

TABLE II
Closely Related Results on Survivable Relay Node Placement

source	connectivity	R vs r	$\mathcal{B} \neq \emptyset$	constraints	approx ratio
[2]	k	$R = r$			$\mathcal{O}(1)$
[15]	2	$R = r$			10
[34]	2	$R \geq r$			14
[34]	2	$R \geq r$	$\surd$		16
this	2	$R \geq r$		$\surd$	**9**
this	2	$R \geq r$	$\surd$	$\surd$	**10**

The problem we are studying here is closely related to the $\{0, 1, 2\}$-*survivable network design problem* (**SNDP**) defined in Definition 4.3. The **SNDP** is known to be NP-hard [26], [27], but admits several polynomial time approximation algorithms [7], [26]. Our approximation algorithms for **RNPs-P** rely on solving instances of **SNDP**.

Definition 4.3: Let $G = (V, E)$ be an undirected graph with nonnegative weights on all edges $e \in E$. For each pair of nodes $u, v \in V$, there is a connectivity requirement $c(u, v) \in \{0, 1, 2\}$. The $\{0, 1, 2\}$-*survivable network design problem* (**SNDP**) asks for a minimum weight subgraph H of G such that for any two nodes $u, v \in V$, H contains at least $c(u, v)$ node-disjoint paths between u and v.

Since the **RNPc-P** problem studied in Section III (which only requires connectivity) is NP-hard, it is natural to believe that the **RNPs-P** problem is also NP-hard. Instead of searching for a hardness proof of the problem, we concentrate on the design and analysis of polynomial time approximation algorithms that have small approximation ratios.

We present a general framework of efficient approximation algorithms, based on approximation algorithms for **SNDP**. In particular, we show that by using the best-known approximation algorithm for **SNDP** [7], our framework becomes an **8**-approximation algorithm for the general **RNPs-P** problem, and a **7**-approximation algorithm for the special **RNPs-P** problem where $\mathcal{B} = \emptyset$. Table II lists the most closely related results on this topic.

B. A Framework of Efficient Approximation Algorithms

In this section, we present a framework of polynomial time approximation algorithms for **RNPs-P**. Our framework is based on polynomial time approximation algorithms for $\{0, 1, 2\}$-**SNDP**. Our framework for **RNPs-P** is presented as Algorithm 2.

Algorithm 2 Approximation for**RNPs-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$

Input : $R \geq r > 0$, set of SNs $\mathcal{X}$, set of BSs $\mathcal{B}$, set of candidate locations of RNs $\mathcal{Z}$, and an approximation algorithm $\mathcal{A}$ for the $\{0, 1, 2\}$-SNDP.

Output : A $\mathcal{F}$-RNPs for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ given by $\mathcal{Y}_{\mathcal{A}} \subseteq \mathcal{Z}$.

1: Construct $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$.

2: **if** the nodes in $\mathcal{B} \cup \mathcal{X}$ are not in a single biconnected component of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$

3: The **RNPs-P** problem does not have a feasible solution.

Stop.

4: **end if**

5: Assign edge weights to the edges in $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ as in Definition 2.2.

6: Assign connectivity requirements between every pair of nodes in G in the following way. Let u and v be two nodes. If neither of them is in $\mathcal{Z}$, set $c(u, v) = 2$. Otherwise, set $c(u, v) = 0$.

7: Apply the polynomial time β-approximation algorithm $\mathcal{A}$ to compute a low weight biconnected subgraph $\mathcal{H}_{\mathcal{A}}$ of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ which meets the connectivity requirement specified in the previous step of this algorithm.

8: Output $\mathcal{Y}_{\mathcal{A}} = \mathcal{Z} \cap V(\mathcal{H}_{\mathcal{A}})$.

The major steps of our scheme are as follows. First, we construct $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, as if we were placing a RN at every candidate location in $\mathcal{Z}$. This is accomplished in Line 1 of the algorithm. The given instance of the problem has a feasible solution if and only if all of the BSs and SNs are in the same biconnected component of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. We can compute all of the biconnected components of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ in linear time using depth first search [6]. This is accomplished in Lines 2–4 of the algorithm. Next, we assign nonnegative integer weights to the edges of the HCG as in Definition 2.2. This is accomplished in Line 5 of the algorithm. In Line 6, we construct an instance of the $\{0,\ 1,\ 2\}$-**SNDP** problem. Then, we apply algorithm $\mathcal{A}$ to compute a low-weight biconnected subgraph $\mathcal{H}_{\mathcal{A}}$ of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, spanning all nodes in $\mathcal{B} \cup \mathcal{X}$. This is accomplished in Line 7 of the algorithm. Finally, in Line 8, we identify the locations to place the RNs.

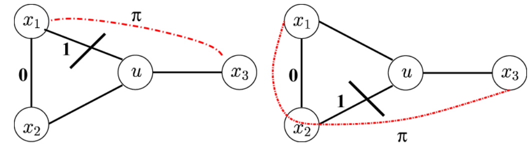
Theorem 4.1: Algorithm 2 has a worst-case running time bounded by $O(|\mathcal{B} \cup \mathcal{X} \cup \mathcal{Z}|^2 + T(\mathcal{A}))$, where $T(\mathcal{A})$ is the time complexity of the approximation algorithm $\mathcal{A}$ used for approximating $\{0,\ 1,\ 2\}$-**SNDP**. Furthermore, we have the following:

- **RNPs-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ has a feasible solution if and only if HCG $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ has a biconnected component that contains all nodes in $\mathcal{B} \cup \mathcal{X}$.
- When **RNPs-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ has a feasible solution, Algorithm 2 guarantees computing a feasible solution that uses no more than $\frac{\beta}{2}(\Delta(\mathcal{H}_{\text{opt}}) + 4)$ times the number of RNs required in an optimal solution $\mathcal{Y}_{\text{opt}}$, where $\mathcal{H}_{\text{opt}}$ is a minimum-weight biconnected subgraph of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{\text{opt}})$ that spans all nodes in $\mathcal{B} \cup \mathcal{X} \cup \mathcal{Y}_{\text{opt}}$, and β is the approximation ratio of $\mathcal{A}$. □

Proof: Let $\mathcal{H}_{\min}$ be an optimal solution of the $\{0,\ 1,\ 2\}$-**SNDP** instance. Since $\mathcal{H}_{\text{opt}}$ is a feasible solution to $\{0,\ 1,\ 2\}$-**SNDP**, and $\mathcal{A}$ is a β-approximation algorithm for $\{0,\ 1,\ 2\}$-**SNDP**, we have

$$w(\mathcal{H}_{\mathcal{A}}) \leq \beta \cdot w(\mathcal{H}_{\min}) \leq \beta \cdot w(\mathcal{H}_{\text{opt}}). \tag{4.1}$$

We need to find an upper bound on $w(\mathcal{H}_{\text{opt}})$ using a function of $|\mathcal{Y}_{\text{opt}}|$. Let $w_2(\mathcal{H}_{\text{opt}})$ denote the total weights of the 2-weight edges in $\mathcal{H}_{\text{opt}}$, and let $w_1(\mathcal{H}_{\text{opt}})$ denote



(a) (b)

Fig. 3. Proof of Lemma 4.1: $\Delta_s(\mathcal{H}_{\text{opt}}) \leq 5$ (a) Delete (x_1, u) (b) Delete edge (x_2, u).

the total weights of the 1-weight edges in $\mathcal{H}_{\text{opt}}$. We have $w(\mathcal{H}_{\text{opt}}) = w_2(\mathcal{H}_{\text{opt}}) + w_1(\mathcal{H}_{\text{opt}})$. Since each RN in $\mathcal{H}_{\text{opt}}$ is incident with at most $\Delta(\mathcal{H}_{\text{opt}})$ 1-weight edges in $\mathcal{H}_{\text{opt}}$, we have

$$w_1(\mathcal{H}_{\text{opt}}) \leq |\mathcal{Y}_{\text{opt}}| \cdot \Delta(\mathcal{H}_{\text{opt}}). \tag{4.2}$$

Applying Lemma 2.3 to each of the connected components of the subgraph of $\mathcal{H}_{\text{opt}}$ induced by all the 2-weight edges, we have

$$w_2(\mathcal{H}_{\text{opt}}) \leq 2 \cdot (2|\mathcal{Y}_{\text{opt}}| - 1). \tag{4.3}$$

It follows from Lemma 2.1 that

$$s(\mathcal{H}_{\mathcal{A}}) \leq \frac{1}{2} w(\mathcal{H}_{\mathcal{A}}) \leq \frac{\beta}{2} w(\mathcal{H}_{\min}) \tag{4.4}$$

$$\leq \frac{\beta}{2} w(\mathcal{H}_{\text{opt}}) \leq \frac{\beta}{2}(4 + \Delta(\mathcal{H}_{\text{opt}}))|\mathcal{Y}_{\text{opt}}|. \tag{4.5}$$

This proves the theorem. ■

There are several choices of the approximation algorithm $\mathcal{A}$ for $\{0,\ 1,\ 2\}$-**SNDP**. For example, if we use the algorithm of [7], the corresponding approximation ratio is $\beta = 2$. If we use the algorithm of [26], the corresponding approximation ratio is $\beta = 3$. Next, we will prove a bound for $\Delta(\mathcal{H}_{\text{opt}})$.

Lemma 4.1: Let $\mathcal{Y}_{\text{opt}}$ be an optimal solution to **RNPs-P** $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{\text{opt}})$. Let $\mathcal{H}_{\text{opt}}$ be a minimum-weight biconnected subgraph of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{\text{opt}})$ spanning all nodes in the graph. Then, $\Delta_s(\mathcal{H}_{\text{opt}}) \leq 5$, $\Delta_b(\mathcal{H}_{\text{opt}}) \leq 1$. □

Proof: We prove this by contradiction. Assume that RN u is connected to six sensor nodes $x_1, x_2, \ldots, x_6$ in $\mathcal{H}_{\text{opt}}$. Without loss of generality, assume that $\angle x_1 u x_2 \leq 60^o$. Since $d(u, x_1) \leq r$, $d(u, x_2) \leq r$ and $\angle x_1 u x_2 \leq 60^o$, we have $d(x_1, x_2) \leq r$. Therefore, (x_1, x_2) is an edge in $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{\text{opt}})$. Since the weight of (x_1, x_2) is 0, we can assume that $(x_1, x_2) \in \mathcal{H}_{\text{opt}}$.

Since $\mathcal{H}_{\text{opt}}$ is biconnected, it contains an x_1–x_3 path π that does not go through u. If path π does not go through node x_2 [as shown in Fig. 3(a)], $\mathcal{H}_{\text{opt}}$ contains a cycle (the edges (x_1, x_2), (x_2, u), (u, x_3) concatenated with the path π) and one of its chords (x_1, u). Deleting the chord (x_1, u) from $\mathcal{H}_{\text{opt}}$ will reduce its weight without destroying its biconnectivity [31]. This contradicts the minimum weight property of $\mathcal{H}_{\text{opt}}$. If path π goes through node x_2 [as shown in Fig. 3(b)], $\mathcal{H}_{\text{opt}}$ contains a cycle (the edge (x_1, u) concatenated with the path π) and one of its chords (x_2, u). Deleting the chord (x_2, u) from $\mathcal{H}_{\text{opt}}$ will reduce its weight without destroying its biconnectivity [31]. This again contradicts the minimum weight property of $\mathcal{H}_{\text{opt}}$.

Now, assume that RN u is connected to two BSs b_1 and b_2 in $\mathcal{H}_{\text{opt}}$. Since $\mathcal{Y}_{\text{opt}}$ is an optimal solution, u is connected to a SN

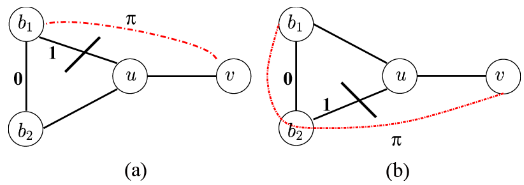


Fig. 4. Proof of Lemma 4.1: $\Delta_b(\mathcal{H}_{\rm opt}) \le 1$ (a) Delete edge (b_1, u) (b) Delete edge (b_2, u).

or another RN v in $\mathcal{H}_{\rm opt}$. Since the weight of (b_1, b_2) is 0, we can assume that $(b_1, b_2) \in \mathcal{H}_{\rm opt}$.

Since $\mathcal{H}_{\rm opt}$ is biconnected, it contains a b_1–v path π that does not go through u. If path π does not go through node b_2 [as shown in Fig. 4(a)], $\mathcal{H}$ contains a cycle (the edges (b_1, b_2), (b_2, u), (u, v) concatenated with the path π) and one of its chords (b_1, u) that has a weight of 1. Deleting the chord (x_1, u) from $\mathcal{H}_{\rm opt}$ will reduce its weight without destroying its biconnectivity [31]. This contradicts the minimum weight property of $\mathcal{H}_{\rm opt}$. If path π goes through node b_2 [as shown in Fig. 4(b)], $\mathcal{H}$ contains a cycle (the edge (b_1, u) concatenated with the path π) and one of its chords (b_2, u). This again contradicts the minimum weight property of $\mathcal{H}_{\rm opt}$. ■

Corollary 4.1: The general **RNPs-P** problem has a **10**-approximation algorithm with a polynomial running time. The special **RNPs-P** problem, where $\mathcal{B} = \emptyset$, has a **9**-approximation algorithm with a polynomial running time. □

Proof: This is achieved by choosing $\mathcal{A}$ as the 2-approximation algorithm of Fleischer [7]. ■

Corollary 4.2: The general **RNPs-P** problem has a **15**-approximation algorithm with a running time of $O(|V|^3 + |E| \cdot |V| \cdot \alpha(|V|))$, where V and E are the node set and edge set of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, and $\alpha(\cdot)$ is the inverse Ackermann function [6]. The special case of **RNPs-P**, where $\mathcal{B} = \emptyset$, has a **13.5**-approximation algorithm with a running time of $O(|V|^3 + |E| \cdot |V| \cdot \alpha(|V|))$. □

Proof: This is achieved by choosing $\mathcal{A}$ as the 3-approximation algorithm of Ravi and Williamson for the $\{0, 1, 2\}$-**SNDP** problem [26], [27]. ■

V. Efficiently Computable Lower Bounds

In order to evaluate the performance of the approximation algorithms that we developed in the previous two sections, we need to compare the approximate solutions with the optimal solutions. The lack of efficient algorithms for computing optimal solutions for **RNPc-P** and **RNPs-P** presents a challenge for the necessary performance study of our approximation algorithms. Since **RNPc-P** is known to be NP-hard, and **RNPs-P** is believed to be NP-hard, it is unlikely that one will be able to compute optimal solutions to these problems in a reasonable amount of time, unless the input size of the instances is very small. In this section, we present a unified linear programming (LP) formulation *that can efficiently compute lower bounds for both* **RNPc-P** *and* **RNPs-P**. These lowers bounds will be used to study the performance of our approximation algorithms in the next section. We denote the LP formulation by $\mathrm{LP}(\mathbf{f})$, where $\mathbf{f} = 1$ corresponds to the case of **RNPc-P** and $\mathbf{f} = 2$ corresponds to the case of **RNPs-P**.

TABLE III
Notation Used in the LP Formulation

$\mathcal{B}$	: the set of base stations
$\mathcal{X}$	: the set of sensor nodes
$\mathcal{Z}$	: the set of candidate locations for relay nodes
HCG	: the hybrid communication graph $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$
$\mathbf{f}$	: $\mathbf{f} = 1$ for connectivity and $\mathbf{f} = 2$ for survivability
t	: base station chosen as the common sink, $t = b_1$
$\mathcal{B}'$	: the set of base stations excluding b_1
f_{uvx}	: variable defined for every $x \in \mathcal{X}$ and every $(u, v) \in \mathsf{HCG}$, denoting *type-x* flow on edge $(u, v) \in \mathsf{HCG}$
r_{zx}	: variable defined for every $x \in \mathcal{X}$ and every $z \in \mathcal{Z}$, denoting total flow of *type-x* into node $z \in \mathcal{Z}$
r_{bx}	: variable defined for every $x \in \mathcal{X}$ and every $b \in \mathcal{B}'$, denoting total flow of *type-x* into node $b \in \mathcal{B}'$
f_z	: variable defined for every every $z \in \mathcal{Z}$, denoting maximum flow of any *type* into node $z \in \mathcal{Z}$

Our LP formulation is based on multicommodity flow packing [25] defined on the HCG. We arbitrarily pick one of the base stations (say b_1) as the common sink and route a flow of *type-x* and *value-*$\mathbf{f}$ from node x to this common sink for each sensor node $x \in \mathcal{X}$. This *type-x* flow is distributed on the edges of the HCG (in the variables f_{uvx} for $(u, v) \in \mathsf{HCG}$) and routed toward the common sink node. For each $x \in \mathcal{X}$ and each $z \in \mathcal{Z}$, the amount of *type-x* flow going through node z (denoted by the variable r_{zx}) cannot exceed 1. For each $x \in \mathcal{X}$ and each $b \in \mathcal{B}$ other than the common sink, the amount of *type-x* flow going through node b (denoted by the variable r_{zx}) cannot exceed 1. For each $z \in \mathcal{X}$, the *maximum amount of flow of any type* going through node z is denoted by the variable f_z. The objective of the LP is to minimize $\sum_{z \in \mathcal{Z}} f_z$ subject to flow conservation constraints in addition to the above constraints. When the variables $f_{uvx}, r_{zx}, r_{bx}, f_z$ are restricted to 0 and 1 values, $\mathrm{LP}(\mathbf{f})$ becomes an integer LP problem, which we denote by $\mathrm{ILP}(\mathbf{f})$. A feasible solution to $\mathrm{ILP}(1)$ corresponds to a $\mathcal{F}$-RNPc where there is a relay node placed at $z \in \mathcal{Z}$ iff $f_z = 1$ in the feasible solution, as the solution guarantees that there is a path connecting sensor node x and base station b_1 for each sensor node $x \in \mathcal{Z}$. Similarly, a feasible solution to $\mathrm{ILP}(2)$ corresponds to a $\mathcal{F}$-RNPs where there is a relay node placed at $z \in \mathcal{Z}$ iff $f_z = 1$ in the feasible solution, as the solution guarantees that there are two node-disjoint paths connecting sensor node x and base station b_1 for each sensor node $x \in \mathcal{Z}$. Therefore, the optimal objective function values of $\mathrm{ILP}(1)$ and $\mathrm{ILP}(2)$ are the numbers of relay nodes required in the optimal solutions to **RNPc-P** and **RNPs-P**, respectively, where there is a relay node at $z \in \mathcal{Z}$ iff $f_z = 1$ in the optimal solution. We list the notations used in the LP formulation in Table III and present the LP formulation as $\mathrm{LP}(\mathbf{f})$. See equations (5.1)–(5.11) at the bottom of the next page.

Constraint (5.2) ensures that *the net flow of type-x out of node x is* $\mathbf{f}$. Constraint (5.3) ensures that *the net flow of type-x into node t is* $\mathbf{f}$. Constraints (5.4) and (5.5) ensure that for each $z \in \mathcal{Z}$ and $x \in \mathcal{X}$, *the total flow of type-x into node z is r_{zx} and that the total flow of type-x out of node z is also r_{zx}*. Constraints (5.6) and (5.7) ensure that for each $b \in \mathcal{B}'$ and $x \in \mathcal{X}$, *the total flow of type-x into node b is r_{bx} and that the total flow of type-x out of node b is also r_{bx}*. Constraint (5.8) ensures that *the flow of type-x on each link is a real number between* 0 *and* 1.

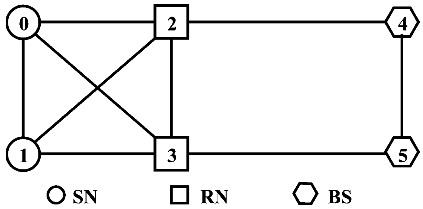


Fig. 5. Example network for the LP formulation illustration.

Constraint (5.9), together with Constraints (5.4) and (5.5), ensures that for each $x \in \mathcal{X}$ and $z \in \mathcal{Z}$, *the total flow of type-x into node z is a real number between* 0 *and* and 1, which equals *the total flow of type-x out of node z*. Constraint (5.10), together with Constraints (5.6) and (5.7), ensures that for each $x \in \mathcal{X}$ and $b \in \mathcal{B}'$, *the total flow of type-x into node b is a real number between* 0 *and* and 1, which equals *the total flow of type-x out of node b*. Constraint (5.11) defines the flow packing, i.e., *for each* $z \in \mathcal{Z}$, f_z *is the maximum usage of node z among all types of flows*. The objective function (5.1) to be minimized is the summation of the maximum usages over all nodes $z \in \mathcal{Z}$.

Fig. 5 shows a small input instance that we shall use to illustrate the LP formulation of **RNPc-P** and **RNPc-P**. The instance consists of two SNs 0 and 1, two RNs 2 and 3, and two BSs 4 and 5. We pick $t = 5$ as the common sink node for the LP formulations. The corresponding LP formulations for **RNPc-P** ($\mathbf{f} = 1$) and **RNPs-P** ($\mathbf{f} = 2$) is shown in Fig. 6.

Next, we will prove that the solution to LP($\mathbf{f}$) leads to a lower bound on the optimal solution for **RNPc-P** (with $\mathbf{f} = 1$) and for **RNPs-P** (with $\mathbf{f} = 2$).

Theorem 5.1: Let F_1 be the objective function value of LP($\mathbf{1}$), and F_2 be the objective function value of LP($\mathbf{2}$). Then, F_1 is a lower bound on the optimal value of **RNPc-P**, and F_2 is a lower bound on the optimal value of **RNPs-P**. □

$$
\begin{aligned}
&\min f_2 + f_3 \\
&\text{s.t.}\quad \underline{\text{Constraints from (5.2):}} \\
&\qquad f_{010} + f_{020} + f_{030} - f_{100} - f_{200} - f_{300} = \mathbf{f} \\
&\qquad f_{101} + f_{121} + f_{131} - f_{011} - f_{211} - f_{311} = \mathbf{f} \\
&\qquad \underline{\text{Constraints from (5.3):}} \\
&\qquad f_{350} + f_{450} - f_{530} - f_{540} = \mathbf{f} \\
&\qquad f_{351} + f_{451} - f_{531} - f_{541} = \mathbf{f} \\
&\qquad \underline{\text{Constraints from (5.4):}} \\
&\qquad f_{020} + f_{120} + f_{320} + f_{420} = r_{20} \\
&\qquad f_{021} + f_{121} + f_{321} + f_{421} = r_{21} \\
&\qquad f_{030} + f_{130} + f_{230} + f_{530} = r_{30} \\
&\qquad f_{031} + f_{131} + f_{231} + f_{531} = r_{31} \\
&\qquad \underline{\text{Constraints from (5.5):}} \\
&\qquad f_{200} + f_{210} + f_{230} + f_{240} = r_{20} \\
&\qquad f_{201} + f_{211} + f_{231} + f_{241} = r_{21} \\
&\qquad f_{300} + f_{310} + f_{320} + f_{350} = r_{30} \\
&\qquad f_{301} + f_{311} + f_{321} + f_{351} = r_{31} \\
&\qquad \underline{\text{Constraints from (5.6):}} \\
&\qquad f_{240} + f_{540} = r_{40} \\
&\qquad f_{241} + f_{541} = r_{41} \\
&\qquad \underline{\text{Constraints from (5.7):}} \\
&\qquad f_{420} + f_{450} = r_{40} \\
&\qquad f_{421} + f_{451} = r_{41} \\
&\qquad \underline{\text{Bound constraints from (5.8) through (5.11):}} \\
&\qquad 0 \le f_{uvx} \le 1, \forall (u,v) \in \mathsf{HCG}, \forall x \in \{0,1\} \\
&\qquad 0 \le r_{ux} \le 1, \forall u \in \{2,3,4\}, \forall x \in \{0,1\} \\
&\qquad r_{zx} \le f_z, \forall z \in \{2,3\}, \forall x \in \{0,1\}
\end{aligned}
$$

Fig. 6. LP formulation for the illustrative deployment setup.

$$\mathrm{LP}(\mathbf{f}) : \min \quad \sum_{z \in \mathcal{Z}} f_z, \text{ over variables } f_{uvx}, r_{zx}, r_{bx}, f_z \tag{5.1}$$

$$\text{s.t.} \quad \sum_{(x,v) \in \mathsf{HCG}} f_{xvx} - \sum_{(u,x) \in \mathsf{HCG}} f_{uxx} = \mathbf{f}, \quad \forall x \in \mathcal{X} \tag{5.2}$$

$$\sum_{(u,t) \in \mathsf{HCG}} f_{utx} - \sum_{(t,v) \in \mathsf{HCG}} f_{tvx} = \mathbf{f}, \quad \forall x \in \mathcal{X} \tag{5.3}$$

$$\sum_{(u,z) \in \mathsf{HCG}} f_{uzx} = r_{zx}, \quad \forall z \in \mathcal{Z}, \quad \forall x \in \mathcal{X} \tag{5.4}$$

$$\sum_{(z,v) \in \mathsf{HCG}} f_{zvx} = r_{zx}, \quad \forall z \in \mathcal{Z}, \quad \forall x \in \mathcal{X} \tag{5.5}$$

$$\sum_{(u,b) \in \mathsf{HCG}} f_{ubx} = r_{bx}, \quad \forall b \in \mathcal{B}', \quad \forall x \in \mathcal{X} \tag{5.6}$$

$$\sum_{(b,v) \in \mathsf{HCG}} f_{bvx} = r_{bx}, \quad \forall b \in \mathcal{B}', \quad \forall x \in \mathcal{X} \tag{5.7}$$

$$f_{uvx} \in [0,1], \quad \forall\, x \in \mathcal{X}, \quad \forall\, (u,v) \in \mathsf{HCG} \tag{5.8}$$

$$r_{zx} \in [0,1], \quad \forall\, z \in \mathcal{Z}, \quad \forall\, x \in \mathcal{X} \tag{5.9}$$

$$r_{bx} \in [0,1], \quad \forall\, b \in \mathcal{B}', \quad \forall\, x \in \mathcal{X} \tag{5.10}$$

$$r_{zx} \le f_z, \quad \forall\, z \in \mathcal{Z}, \quad \forall\, x \in \mathcal{X} \tag{5.11}$$

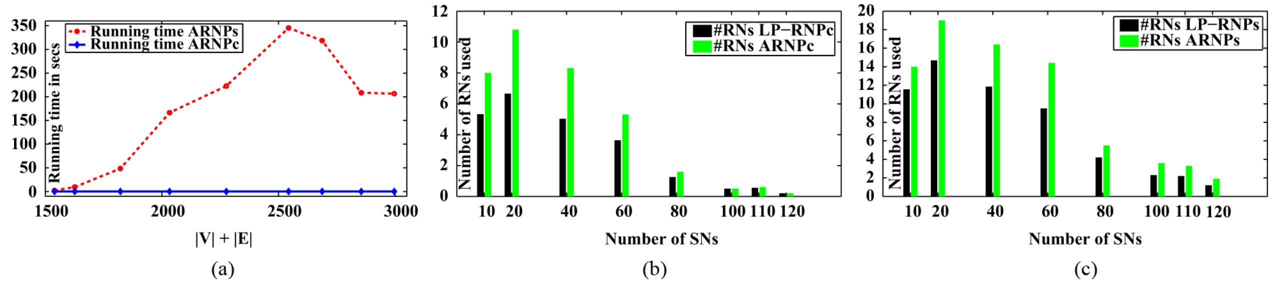


Fig. 7. Results with increasing density: 100×100 playing field; $|\mathcal{Z}| = 121$; $|\mathcal{B} \cup \mathcal{X}| = 10, 20, 40, 60, 80, 100, 110, 120, 130$. (a) Running times of ARNPc and ARNPs. (b) Number of RNs used by ARNPc, LP-RNPc. (c) Number of RNs used by ARNPs, LP-RNPs.

Proof: Let an instance for the relay node placement problem be given by $R \geq r > 0$, sensors $\mathcal{X}$, base stations $\mathcal{B}$, and candidate locations for relay nodes $\mathcal{Z}$. We arbitrarily pick $t = b_1 \in \mathcal{B}$ as the common sink node.

Let $\mathcal{Y}_1 \subseteq \mathcal{Z}$ be a $\mathcal{F}$-RNPc for the given instance. Then, the hybrid communication graph $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_1)$ is connected. Therefore, for each $x \in \mathcal{X}$, there is an x–t path in $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_1)$. We can use this path to define the type-x flow of value-1 in $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. Also, for each $z \in \mathcal{Z}$, set $f_z = 1$ if $z \in \mathcal{Y}_1$, and $f_z = 0$ otherwise. Then, we have a feasible solution for LP(**1**) whose objective function value is equal to $|\mathcal{Y}_1|$. This proves that F_1 is a lower bound for the optimal solution of **RNPc-P**.

Let $\mathcal{Y}_2 \subseteq \mathcal{Z}$ be a $\mathcal{F}$-RNPs for the given instance. Then the hybrid communication graph $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_2)$ is 2-connected. Therefore, for each $x \in \mathcal{X}$, there exists a pair of node-disjoint $x - t$ paths in $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_2)$. We can use this pair of paths to define the type-x flow of value-2 in $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. Also, for each $z \in \mathcal{Z}$, set $f_z = 1$ if $z \in \mathcal{Y}_2$, and $f_z = 0$ otherwise. Then, we have a feasible solution for LP(**2**) whose objective function value is equal to $|\mathcal{Y}_2|$. This proves that F_2 is a lower bound for the optimal solution of **RNPs-P**. ■

Note that LP(**f**) can be solved in polynomial time, which can provide a lower bound on the optimal solutions for **RNPc-P** (with $\mathbf{f} = 1$) and for **RNPs-P** (with $\mathbf{f} = 2$). If an approximate solution produced by an approximation algorithm to **RNPc-P** (**RNPs-P**, respectively) is within a factor of α from its lower bound, it is guaranteed to be within a factor of α from its optimal solution. We will use these efficiently computable lower bounds in our performance studies in the next section.

VI. Numerical Results

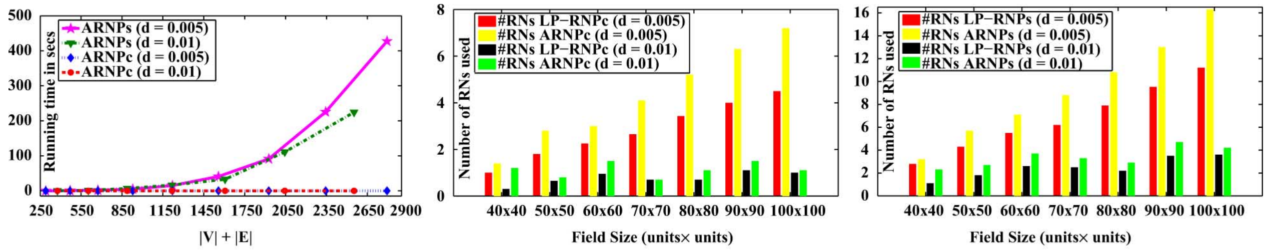
To verify the effectiveness of the frameworks presented in this paper, we have implemented our approximation algorithms for both **RNPc-P** and **RNPs-P**, tested them on a variety of test problems, and compared the solutions obtained by our approximation algorithms with the corresponding lower bounds computed using the LP formulations presented in Section V. We have studied both the running times (for scalability) and the number of relay nodes required (for performance of approximation). The numerical results show that our theoretical analyses are quite conservative. In all cases, the number of relay nodes required in the solutions obtained by our approximation algorithms is very close to the corresponding theoretical lower bounds. This indicates that our approximation algorithms are very effective in practice.

The algorithms implemented are: 1) ARNPc, which is Algorithm 1 with $\mathcal{A}$ being the MST based 2-approximation in [17] for **STP** (simpler than the algorithm in [28]); 2) ARNPs, which is Algorithm 2 with $\mathcal{A}$ being the sequential maximum-flow-based 3-approximation in [26] for $\{0, 1, 2\}$-**SNDP** (simpler than the algorithm in [7]); 3) LP-RNPc, which solves LP(**1**) using ILOG CPLEX [35]; and 4) LP-RNPs, which solves LP(**2**) using ILOG CPLEX [35]. The tests were run on a 2.4-GHz Linux PC with 1 GB of memory.

As in [15] and [34], the SNs $\mathcal{X}$ were randomly distributed in a square playing field. Two base stations were randomly deployed in the field. We used both regular grid points and randomly generated points as the candidate locations for the relay nodes. For both setups, we set $r = 15$ and $R = 30$.

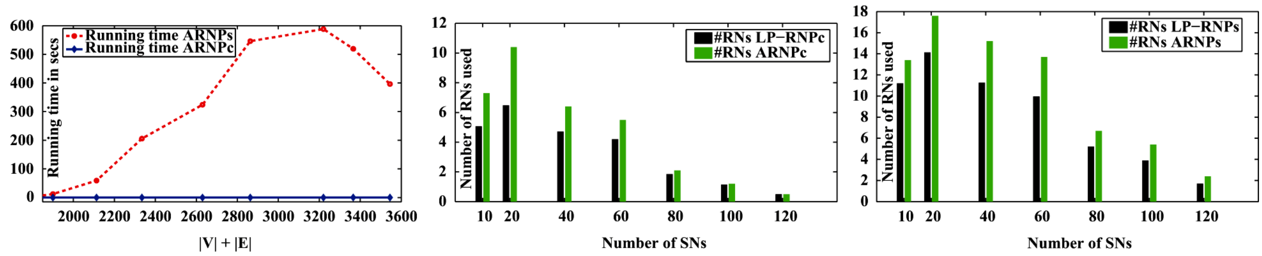
We first present the results for the case where the candidate locations are regular grid points. For this setting, we chose the grid to be a square grid since it is the most commonly used deployment. The playing field consists of $K \times K$ small squares each of side 10, with the $(K+1)^2$ grid points as $\mathcal{Z}$. We studied two separate settings: the case where the density of the SNs in the field increases and the case where the density is constant. We define the density as the ratio between the number of SNs in the field and the area of the field. For the increasing density case, we chose a constant field size of 100×100 square units. For the constant density case, we let the size of the playing field increase with the number of SNs.

In the case with increasing density, the number of candidate RN locations was 121. The number of SNs was varied from 10 to 120. For each setting, the SNs were deployed randomly in the field. The results were averaged over 10 test cases. Fig. 7(a) shows the running time of ARNPs and ARNPc. The x-axis is the sum of the average number of edges and nodes in the HCG (as the running time of the algorithm depends on both $|E|$ and $|V|$), and the y-axis is the running time in seconds. The solid line shows the running time of ARNPc, which is less than 1 s in all cases. This is because of the $\mathcal{O}(n \log n)$ complexity of ARNPc, which is fairly small. The dashed line shows the running time of ARNPs. The running time of ARNPs increases with the increase of the number of SNs and decreases after a certain threshold, as shown in the figure. This is expected because the $\{0, 1, 2\}$-**SNDP** algorithm requires computation of the maximum flows for every pair of SNs in the



(a) (b) (c)

Fig. 8. Results with constant density: seven different playing fields, from 40 × 40 to 100 × 100; two density values, $d_1 = 0.005$ and $d_2 = 0.01$. (a) Running times of ARNPc and ARNPs. (b) Number of RNs used by ARNPc, LP-RNPc. (c) Number of RNs used by ARNPs, LP-RNPs.



(a) (b) (c)

Fig. 9. Random grid—Results with increasing density: 100 × 100 playing field; $|\mathcal{Z}| = 121$; $|\mathcal{B} \cup \mathcal{X}| = 10, 20, 40, 60, 80, 100, 110, 120, 130$. (a) Running times of ARNPc and ARNPs. (b) Number of RNs used by ARNPc, LP-RNPc. (c) Number of RNs used by ARNPs, LP-RNPs.

network that are not biconnected yet. If the number of SNs increases beyond the threshold, the biconnectivity among the SNs also increases correspondingly. This increased biconnectivity reduces the number of maximum flow computations required, resulting in a decrease in the running time. From the figures, we conclude that ARNPc is has a fast running time, whereas ARNPs requires increasing computation with increasing values of $|V| + |E|$ up to a threshold beyond which the computation time decreases. Fig. 7(b) and (c) show the average number of RNs required by ARNPc and ARNPs, respectively, and that required by LP-RNPc and LP-RNPs, respectively. The number of RNs required decreases with an increase in the number of SNs. This is because with an increase in the number of SNs, they are able to satisfy the connectivity and survivability requirements with the help of less number of RNs. From the figures, we can conclude that in a network where the SNs are sparsely deployed, use of an effective algorithm is essential for efficient relay nodes placement. With increasing density, the number of relay nodes required is much lesser, hence the effectiveness of the algorithm is not as crucial. *We note that in all cases, the number of* RNs *required by our approximation algorithms is no worse than twice that of the number obtained from solving the corresponding LPs*. This indicates that our approximation algorithms perform very well.

For the case of constant density, we studied two subcases: one with density $d_1 = 0.005$ and the other with density $d_2 = 0.01$. For each density value, we used seven different numbers of SNs. The field sizes were chosen to be $40 \times 40, \ldots, 100 \times 100$, with the number of SNs ranging from 8 to 50 for d_1, and 16 to 100 for d_2, deployed randomly in the field. The result of each configuration was averaged over 10 test cases. Fig. 8(a) shows the running times of ARNPc and ARNPs. For both densities, ARNPc has running time less than 1 s because of its small time complexity. On the other hand, the running time of ARNPs is dependent on both the density and the number of SNs in the network. The running time for $d_2 = 0.01$ is lesser than that of $d_1 = 0.005$. This is expected because with the increase in density, more pairs of SNs are already biconnected. Hence, our algorithm runs faster with an increase in density. Fig. 8(b) and 8(c) show the number of RNs required by various algorithms. *Our algorithms perform very well in comparison with the results obtained by solving the LPs. In the worst case, the number of* RNs *obtained from* ARNPc *is four times the number obtained from solving the corresponding LPs*. This is the case where the density is d_2 and the field size is 40×40. *This shows that our theoretical analysis on the approximation ratios of our approximation algorithms is quite conservative. In other words, our approximation algorithms perform quite better than the theoretical approximation ratios indicate*.

Now, we present results where the candidate locations of the RNs were randomly generated. This simulates a random deployment of the RNs, say, from an aircraft or a terrestrial vehicle. In this setup, we studied the settings where the density of the SNs in the field increases and where the density remains constant. The parameters of the simulations are exactly the same barring the fact that the RNs are now randomly deployed in the square field instead of on a square grid. This setup may also be thought of as the placement of the RNs on a random grid. Fig. 9 shows the results for the increasing density case. The trend is similar to the case with square grid. Despite the trend being the same, the value of the running time is higher in the random grid case than in the square grid case. This is because in the random grid case, due to the random deployment of the RNs, a RN chosen at an iteration in the algorithm may only be able to biconnect one SN that needs to be biconnected. This results in more computation to ensure that all the SNs are biconnected, thus increasing

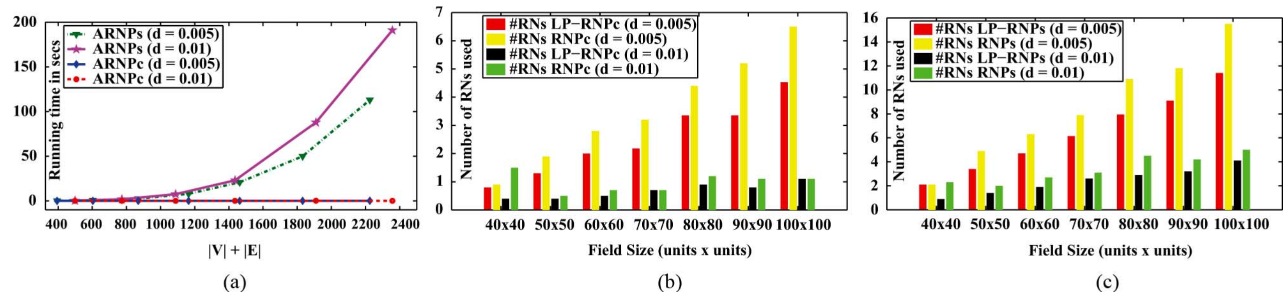


Fig. 10. Random grid—Results with constant density: seven different playing fields, from 40×40 to 100×100; two density values, $d_1 = 0.005$ and $d_2 = 0.01$ (a) Running times of ARNPc and ARNPs (b) Number of RNs used by ARNPc, LP-RNPc (c) Number of RNs used by ARNPs, LP-RNPs.

the running time. Fig. 9(b) and (c) show the number of RNs used for the connectivity and survivability cases, respectively. *The number of* RNs *obtained by our approximation algorithms is within two times the number obtained by the solution of the corresponding LPs*.

Fig. 10 shows the results for the constant density case. The running time of the ARNPc and ARNPs algorithms is shown in Fig. 10(a). In comparison to the running time plot in the square grid case, we can see that the running time of the ARNPs for $d_2 = 0.01$ is higher than the running time for $d_1 = 0.005$. This is due to the random placement of the RNs. *The number of* SNs *when the density is d_2 is twice that when in the density is d_1*. Owing to the random deployment of the RNs, it takes more time to biconnect the SNs in the network. Thus, we can conclude that, in the random grid case, increase in the density of the SNs can increase the running time of ARNPs. Fig. 10(b) shows the number of RNs required when ARNPc and LP-RNPc are used. As expected, the number of RNs required when the density is $d_2 = 0.01$ is lesser than that required when the density is $d_1 = 0.005$. The number of RNs required by ARNPc is less than four times that required by the LP solution in the worst case (40×40 case). Fig. 10(c) shows the number of RNs required for survivability. The number of RNs required by ARNPs is never more than twice the number required by the LP solution. The simulation results show that our approximation algorithms are very effective in solving **RNPc-P** and **RNPs-P**.

VII. Conclusion

In this paper, we have formulated constrained single-tiered relay node placement problems in a heterogeneous wireless sensor network to meet connectivity and survivability requirements. We have discussed the computational complexities of these problems and presented a framework of polynomial time approximation algorithms with $\mathcal{O}(1)$ approximation ratios. To our best knowledge, we are the first to present $\mathcal{O}(1)$ approximation algorithms for the constrained relay node placement problems.

The connectivity requirement in this paper ensures the existence of a bidirectional path between each sensor node and a base station, which supports both broadcast from a base station and data collection to the base stations. A weaker connectivity requirement is one that ensures the existence of a directional path from each sensor node to a base station, which supports data collection only. Obviously, the number of relay nodes required to ensure this weaker connectivity will not exceed the number of relay nodes required to ensure the stronger connectivity studied in this paper. The study of constrained relay node placement under this weaker connectivity requirement may be a direction of future research.

Instead of placing the more powerful and more expensive relay nodes to meet the connectivity or survivability requirement, one may also place the less expensive sensor nodes to meet the connectivity or survivability requirement. This corresponds to the special case of $R = r$. The algorithms studied in this paper apply to this case as well. However, approximation algorithms with smaller approximation ratios may exist for this special case. Therefore, the study of better algorithms for this case is also of interest.

Acknowledgment

The authors wish to thank the associate editor and the anonymous reviewers, whose constructive and insightful comments on an earlier version of this paper have helped to significantly improve the presentation of the paper.

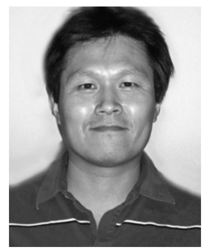
References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, pp. 393–422, 2002.
[2] J. Bredin, E. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance," in *Proc. ACM MobiHoc*, 2005, pp. 309–319.
[3] D. Chen, D. Z. Du, X. D. Hu, G. Lin, L. Wang, and G. Xue, "Approximations for Steiner trees with minimum number of Steiner points," *J. Global Optimization*, vol. 18, pp. 17–33, 2000.
[4] P. Cheng, C. N. Chuah, and X. Liu, "Energy-aware node placement in wireless sensor networks," in *Proc. IEEE Globecom*, 2004, pp. 3210–3214.
[5] X. Cheng, D. Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Netw.*, vol. 14, pp. 347–355, 2008.
[6] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press and McGraw-Hill, 2001.
[7] L. Fleischer, "A 2-Approximation for minimum cost {0, 1, 2} vertex connectivity," in *Proc. IPCO*, 2001, pp. 115–129.
[8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.
[9] G. Gupta and M. Younis, "Fault tolerant clustering of wireless sensor networks," in *Proc. WCNC*, 2003, pp. 1579–1584.
[10] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," in *Proc. IEEE INFOCOM*, 2007, pp. 1649–1657.

[11] B. Hao, J. Tang, and G. Xue, "Fault-tolerant relay node placement in wireless sensor networks: Formulation and approximation," in *Proc. IEEE HPSR*, 2004, pp. 246–250.
[12] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "Prolonging sensor network lifetime with energy provisioning and relay node placement," in *Proc. IEEE Secon*, 2005, pp. 295–304.
[13] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*. Amsterdam, The Netherlands: North Holland, 1992, Annals of Discrete Mathematics.
[14] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM Mobicom*, 2001, pp. 243–254.
[15] A. Kashyap, S. Khuller, and M. Shayman, "Relay placement for higher order connectivity in wireless sensor networks," in *Proc. IEEE INFOCOM*, pp. 1–12.
[16] W.-C. Ke, B.-H. Liu, and M.-J. Tsai, "Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is NP-complete," *IEEE Trans. Comput.*, vol. 56, no. 5, pp. 710–715, May 2007.
[17] L. T. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees," *Acta Informatica*, vol. 15, pp. 141–145, 1981.
[18] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless ad-hoc networks," in *Proc. ACM Mobicom*, 2001, pp. 97–107.
[19] G. Lin and G. Xue, "Steiner tree problem with minimum number of Steiner points and bounded edge-length," *Inf. Process. Lett.*, vol. 69, pp. 53–57, 1999.
[20] H. Liu, P. J. Wan, and X. H. Jia, "Fault-tolerant relay node placement in wireless sensor networks," *LNCS*, vol. 3595, pp. 230–239, 2005.
[21] E. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Trans. Comput.*, vol. 56, no. 1, pp. 134–138, Jan. 2007.
[22] W. Lou, W. Liu, and Y. Fang, "SPREAD: Enhancing data confidentiality in mobile ad hoc networks," in *Proc. IEEE INFOCOM*, 2004, pp. 2404–2413.
[23] S. Misra, D. Hong, G. Xue, and J. Tang, "Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements," in *Proc. IEEE INFOCOM*, 2008, pp. 281–285.
[24] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, "Topology control for wireless sensor networks," in *Proc. ACM Mobicom*, 2003, pp. 286–299.
[25] S. Plotkin, D. Shmoys, and E. Tardos, "Fast approximation algorithms for fractional packing and covering problems," *Math. Oper. Res.*, vol. 20, pp. 257–301, 1995.
[26] R. Ravi and D. P. Williamson, "An approximation algorithm for minimum-cost vertex-connectivity problems," *Algorithmica*, vol. 18, pp. 21–43, 1997.
[27] R. Ravi and D. P. Williamson, "Erratum: An approximation algorithm for minimum-cost vertex-connectivity problems," *Algorithmica*, vol. 34, pp. 98–107, 2002.
[28] G. Robins and A. Zelikovsky, "Tighter bounds for graph Steiner tree approximation," *SIAM J. Discrete Math.*, vol. 19, pp. 122–134, 2005.
[29] A. Srinivas, G. Zussman, and E. Modiano, "Mobile backbone networks-Construction and maintenance," in *Proc. ACM Mobihoc*, 2006, pp. 166–177.
[30] J. Tang, B. Hao, and A. Sen, "Relay node placement in large scale wireless sensor networks," *Comput. Commun.*, vol. 29, pp. 490–501, 2006.
[31] D. B. West, *Introduction to Graph Theory*. Englewood Cliffs, NJ: Prentice Hall, 1996.
[32] K. Xu, H. Hassanein, and G. Takahara, "Relay node deployment strategies in heterogeneous wireless sensor networks: multiple-hop communication case," in *Proc. IEEE Secon*, 2005, pp. 575–585.
[33] O. Younis and S. Fahmy, "Distributed clustering for ad-hoc sensor networks: A hybrid, energy-efficient approach," in *Proc. IEEE INFOCOM*, 2004, pp. 269–640.
[34] W. Zhang, G. Xue, and S. Misra, "Fault-tolerant relay node placement in wireless sensor networks: Problems and algorithms," in *Proc. IEEE INFOCOM*, 2007, pp. 1649–1657.
[35] "ILOG CPLEX: Software for mathematical programming and optimization," ver. 8.1, 2002 [Online]. Available: http://www.ilog.com/products/cplex/

Satyajayant Misra (S'04–M'09) received the integrated M.Sc. (Tech.) information systems and M.Sc. (Hons.) physics degrees from the Birla Institute of Technology and Science, Pilani, India, in June 2003, and the Ph. D. degree in computer science from Arizona State University, Tempe, in 2009.

He is an Assistant Professor with the Department of Computer Science, New Mexico State University, Las Cruces. His research interests include identifying security, privacy, and reliability issues in wireless sensors and ad hoc networks and formulating efficient solutions to handle them.



Seung Don Hong received the B.S. degree in computer science and engineering from Arizona State University, Tempe, in May 2005.

Since then, he has been a Ph.D. student in the Department of Computer Science and Engineering, Arizona State University. His research interests include wireless sensor networks.

Guoliang (Larry) Xue (SM'99) received the B.S. degree in mathematics and the M.S. degree in operations research from Qufu Teachers University, Qufu, China, in 1981 and 1984, respectively, and the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, in 1991.

He is a Full Professor of computer science and engineering with Arizona State University, Tempe, and has held previous positions at Qufu Teachers University; the Army High Performance Computing Research Center, Stanford, CA; and the University of Vermont, Burlington. His research interests include efficient algorithms for optimization problems in networking, with applications to QoS, survivability, security, privacy, and energy efficiency issues in networks. He has published over 160 papers in these areas. His research has been continuously supported by federal agencies including the NSF and ARO.

Prof. Xue has been a Member of the Association for Computing Machinery (ACM) since 1993. He received the Graduate School Doctoral Dissertation Fellowship from the University of Minnesota in 1990, a third prize from the Ministry of Education of P.R. China in 1991, a NSF Research Initiation Award in 1994, and a NSF-ITR Award in 2003. He is an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (TWC), an Area Editor of *Computer Networks* (COMNET), an Editor of *IEEE Network*, and an Associate Editor of the *Journal of Global Optimization*. He has served on the executive/program committees of many conferences, including ACM Mobihoc, IEEE INFOCOM, IEEE Secon, IEEE ICC, and IEEE Globecom. He was a TPC Co-Chair of the IEEE ICC Symposium on Wireless Ad Hoc and Sensor Networks in 2007 and 2009, and a TPC Co-Chair of QShine in 2007. He is serving as a TPC Co-Chair of IEEE INFOCOM 2010.

Jian Tang (S'04–M'08) received the Ph.D. degree in computer science from Arizona State University, Tempe, in 2006.

He is an Assistant Professor with the Department of Computer Science, Montana State University, Bozeman. His research interest is in the area of wireless networking, with emphasis on routing, scheduling, cross-layer optimization, and QoS provisioning. He has published over 40 papers in this area. His research has been supported by the National Science Foundation, U.S. Department of Homeland Security, and Montana State.

Dr. Tang is the recipient of an NSF CAREER award. He has served on the executive or program committees of many international conferences such as IEEE INFOCOM, IEEE ICC, IEEE Globecom, IEEE IWQoS, and IEEE ICCCN.