This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

Constrained Relay Node Placement in Wireless Sensor Networks to Meet Connectivity and Survivability Requirements

Satyajayant Misra, Seung Don Hong, Guoliang Xue, and Jian Tang

***Abstract*— 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 and survivability requirements. In this paper, we study *constrained* versions of the relay node placement problem, where relay nodes can only be placed at a subset of candidate locations. In the connected relay node placement problem, we want to place a minimum number of relay nodes to ensure the connectivity of the sensor nodes and the base stations. In the survivable relay node placement problem, we want to place a minimum number of relay nodes to ensure the biconnectivity of the sensor nodes and the base stations. 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.**

Keywords: **Relay node placement, wireless sensor networks.**

1. Introduction and Motivations

A wireless sensor network (WSN) consists of many low-cost and low-power *sensor nodes* (SNs)[1]. Two fundamental functions of an SN in a WSN are to sense its environment and to transmit sensed information to the base stations (BSs). There has been extensive research on energy aware routing [4, 14, 19, 31], improvement in lifetime [12, 24, 30], and survivability [23]. 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, 21, 22, 30]. These problems are studied under the 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, 28], and Infocom [10, 15, 32].

Relay node placement problems can be classified into either *single-tiered* or *two-tiered* based on the routing structures [11, 12, 22, 24], and into either *connected* or *survivable* based on the connectivity requirements [2, 11, 15, 32]. In single-tiered relay node placement, an SN also forwards packets received from other nodes. In two-tiered relay node placement, an SN forwards its sensed information to an RN or a 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 the SNs and BSs are connected. In survivable relay node placement, we place a small number of RNs to ensure that the SNs and BSs are biconnected.

We first review prior works on single-tiered relay node placement. In our discussions, we will use R and r to denote the communication ranges of RNs and SNs, respectively. We will use $k = 1$ to denote connectivity requirement and use $k \geq 2$ to denote survivability requirement. In 1999, Lin and Xue [20] 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, 21, 22, 28, 32]. 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 above works assume that the transmission range of the RNs is the same as that of the SNs. Lloyd and Xue [22] studied the problem with $R \geq r$ and $k = 1$, proved its NP-hardness, and presented a 7-approximation algorithm. Zhang *et al.* [32] presented a 14-approximation algorithm for $R \geq r$ and $k = 2$.

Motivated by the works [9] and [24] on two-tiered WSNs, Hao *et al.* [11] formulated two-tiered relay node placement problems where each SN has to be within distance r of at least k RNs and the RNs form a k-connected network. Works along this line can be found in [10, 21, 22, 28, 32].

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, 22, 32], the relay nodes are stacked on top of other relay nodes or sensor nodes. In practice, however, there may be a lower bound on the distance between two network nodes, and also forbidden regions where relay nodes cannot be placed. As a first step to solving this challenging problem, we study a *constrained* relay node placement where the RNs can only be placed at a subset of candidate locations.

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.

Satyajayant Misra, Seung Don Hong, and Guoliang Xue are all with the Department of Computer Science and Engineering at Arizona State University, Tempe, AZ 85287. Email: {satyajayant, seung.hong, xue}@asu.edu. This research was supported in part by ARO grant W911NF-04-1-0385 and NSF grants CCF-0431167 and CNS-0721803. The information reported here does not reflect the position or the policy of the federal government.

Jian Tang is with the Department of Computer Science at Montana State University, Bozeman, MT 59717. Email: tang@cs.montana.edu.

In Section 2, we present basic notations and prove a few fundamental lemmas. In Section 3, we study *connected* relay node placement problem. In Section 4, we study *survivable* relay node placement problem. We present numerical results in Section 5 and conclude the paper in Section 6.

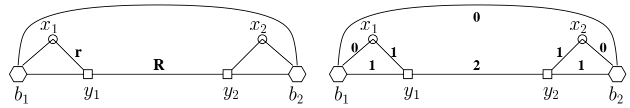
2. Basic Notations and Fundamental Lemmas

We consider a *hybrid wireless sensor network* (HWSN) consisting of *sensor nodes* (SNs), *relay nodes* (RNs), and *base stations* (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 might have to 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, this assumption simplifies notation 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, an SN u can communicate directly with another node v (which could be an SN, an RN or a BS) if and only if $d(u,v) \leq r$. An RN u can communicate directly with another node v (which could be an 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* 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 vertex 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 an RN $y \in \mathcal{Y}$ and a node $z \in \mathcal{B} \cup \mathcal{Y}$ which could be either an RN or a BS, E contains the undirected edge $(y,z) = (z,y)$ if and only if $d(y,z) \leq R$. For an SN $x \in \mathcal{X}$ and a node $z \in \mathcal{B} \cup \mathcal{X} \cup \mathcal{Y}$ which is either an SN, an 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 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.



(a) Illustration of HCG (b) Edge weights in HCG

Fig. 1. (a) shows 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) shows the edge weights, 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.

The hybrid communication graph defines all possible pairwise 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 vertex set of G and $E(G)$ denotes the edge set of G.

Definition 2.2: Let $G = $ 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

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

The *relay size* of G, denoted by $s(G)$, is the number of relay nodes in G, i.e., $s(G) = |V(G) \cap \mathcal{Y}|$. 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

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

Fig. 1(b) illustrates the edge weights of the HCG shown in Fig. 1(a). 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 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 which 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, add a weight of 1 to node v. Let (u,v) be an edge of H where u is an 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 an RN. According to our definition, the weight of this edge is 0. In this case, we add a weight of 0 to node u, 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 which 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(b). 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 which 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.

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 [29]. 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 vertex, but never two. Assume that the union of $H_1, \ldots, H_k$ has p connected components ($1 \leq p \leq k$). Let $V \setminus (H_1 \cup \cdots \cup H_k) = \{v_1, v_2, \ldots, v_q\}$, 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

$$
\begin{aligned}
|E'| &= \sum_{i=1}^{k} |E(H_i)| + (p+q-1) && (2.4)\\
&\leq \sum_{i=1}^{k} (2n_i - 3) + (p+q-1) && (2.5)\\
&= 2|V| - 1 - (k+p+q) \leq 2|V| - 1, && (2.6)
\end{aligned}
$$

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 non-relay 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 [29]. 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.

3. Relay Node Placement to Ensure Connectivity

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

Relay node placement in wireless sensor networks has been studied by many researchers [2, 3, 5, 10, 11, 15, 20, 21, 22, 28, 32]. 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 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. In the next section, we study the problem of placing the minimum number of RNs to ensure network survivability.

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 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 F-RNPc is $|\mathcal{Y}|$. An F-RNPc is said to be a *minimum connected relay node placement* for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ (denoted by M-RNPc) if it has the minimum size among all 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 $\mathsf{RNPc}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, seeks an M-RNPc for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. □

We also study a special case of the RNPc problem where $\mathcal{B} = \emptyset$. Many existing works correspond to this special case [3, 5, 20, 22]. For this special case, our algorithm has a faster running time, and a better approximation ratio.

Computational Complexity:

The RNPc problem is NP-hard, as the authors of [16] have proved that the problem is NP-hard even for the special case where all the nodes are on regular triangular grid points. Therefore we seek efficient algorithms that have provably good performance guarantees.

Our Contributions:

We present a general *framework* of efficient approximation algorithms, 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 [27], our framework becomes a $\mathbf{5.5}$-approximation algorithm for the RNPc problem when $\mathcal{B} = \emptyset$, and a $\mathbf{6.2}$-approximation algorithm for the general RNPc problem. 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 the RNPc problem when $\mathcal{B} = \emptyset$ is the *single tiered relay node placement problem* (1tRNP) studied by Lloyd and Xue [22], where there is no restriction on the locations of the relay nodes. Considering that the best-known approximation algorithm [22] for 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.

TABLE I

CLOSELY RELATED RESULTS ON CONNECTED RELAY NODE PLACEMENT

source	connectivity	R vs r	$\mathcal{B} \neq \emptyset$	constraints	approx ratio
[20]	1	$R = r$			5
[3]	1	$R = r$			3
[5]	1	$R = r$			3
[22]	1	$R \geq r$			7
this	1	$R \geq r$		$\surd$	**5.5**
this	1	$R \geq r$	$\surd$	$\surd$	**6.2**

From a simplicity stand point, we show that by using the minimum spanning tree (MST) based approximation algorithm for STP, our framework becomes a $\mathbf{7}$-approximation algorithm for the RNPc problem when $\mathcal{B} = \emptyset$, and an $\mathbf{8}$-approximation algorithm for the general RNPc problem.

B. A Framework of Efficient Approximation Algorithms

In this section, we present a framework of polynomial time approximation algorithms for the RNPc problem. 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 is presented as Algorithm 1.

The major steps of the algorithm are as follows. First, we construct $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, as if we were placing an RN at every candidate location in $\mathcal{Z}$. This is accomplished in

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

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.

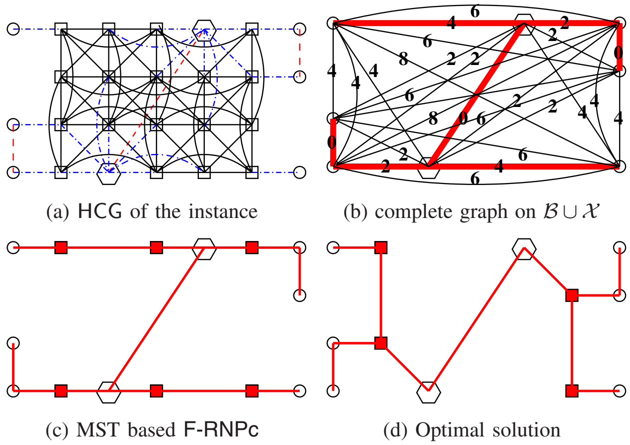
Output: An F-RNPc for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ given by $\mathcal{Y}_\mathcal{A} \subseteq \mathcal{Z}$.

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

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 $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. 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 it is incident with. 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.

The STP problem admits several polynomial time approximation algorithms with small constant approximation ratios. For example, following the ideas of [18], we can construct an edge-weighted complete graph $\mathcal{C}$ on the vertex set $\mathcal{B} \cup \mathcal{X}$, where the weight of an edge connecting two vertices u and v in $\mathcal{C}$ is the length of the shortest u–v path in $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. Computing an MST of $\mathcal{C}$, and replacing each edge in the MST by the corresponding shortest path in $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ leads to a connected subgraph of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ which connects all nodes in $\mathcal{B} \cup \mathcal{X}$ and has a weight no more than twice that of the optimal solution. Other more sophisticated approximation algorithms are also known. For example, with a longer running time (still a polynomial time algorithm) the algorithm of [27] has an approximation ratio of $1 + \frac{\ln 3}{2} \leq 1.55$.

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 dash lines (red color), the 1-weight edges are shown in dash-dot lines (blue color), and the 2-weight edges are shown in solid lines (black color). Fig. 2(b) shows the



(a) HCG of the instance (b) complete graph on $\mathcal{B} \cup \mathcal{X}$

(c) MST based F-RNPc (d) Optimal solution

Fig. 2. (a) The HCG 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 an MST (thick edges). (c) The corresponding F-RNPc, which uses six RNs. (d) The optimal solution, which uses four RNs.

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. An MST of the complete graph is shown in thick (red color) edges. Fig. 2(c) shows the relay node placement corresponding to the MST, which uses six RNs, shown as filled squares (red color). 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:

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

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$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, 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 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 $\mathcal{T}_{opt}$ 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

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

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

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

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

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

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

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

Therefore

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

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

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

This proves the theorem. ■

There are several choices for the approximation algorithm $\mathcal{A}$. For example, if we use the algorithm of [18], the corresponding approximation ratio is $\beta = 2$. If we use the algorithm of [27], the corresponding approximation ratio is $\beta = 1 + \frac{\ln 3}{2} \leq 1.55$. Next we will prove a bound on $\Delta(\mathcal{T}_{opt})$.

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

PROOF. We prove this by contradiction. Assume that in $\mathcal{T}_{opt}$, an 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 $\mathcal{T}_{opt}$ 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 $\mathcal{T}_{opt}$ 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 $w(\mathcal{T}_1) < w(\mathcal{T}_{opt})$, contradicting the assumption that $\mathcal{T}_{opt}$ is a minimum spanning tree. Therefore an RN u cannot be connected to more than five SNs in $\mathcal{T}_{opt}$.

Now assume that a relay node u is connected to two BSs b_1 and b_2 in $\mathcal{T}_{opt}$. Since $\mathcal{T}_{opt}$ is a tree, it does not contain the edge (b_1, b_2). We can replace edge (u, b_1) in $\mathcal{T}_{opt}$ 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 $\mathcal{T}_{opt}$ can be connected to more than one BSs. ■

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

PROOF. According to Robins and A. Zelikovsky [27], 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 $\Delta(\mathcal{T}_{opt})$ bound derived in Lemma 3.1. ■

Corollary 3.2: The general RNPc problem has an $\mathbf{8}$-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 RNPc problem (with $\mathcal{B} = \emptyset$) has a $\mathbf{7}$-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 [18], 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. ■

Note that the $\mathbf{7}$-approximation for the *constrained* problem matches the best-known algorithm for the corresponding *unconstrained* problem [22], while the $\mathbf{5.5}$-approximation for the constrained problem compares favorably with the best-known algorithm for the unconstrained problem [22].

4. Relay Node Placement to Ensure Survivability

In Section 3, 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 which 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, 21, 28, 32]. 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, 21, 28, 32] study relay node placement that ensures 2-connectivity, while the works [2, 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 pre-specified 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 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 F-RNPs is $|\mathcal{Y}|$. An F-RNPs is said to be a *minimum survivable relay node placement* for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ (denoted by M-RNPs) if it has the minimum size among all 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 $\mathsf{RNPs}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, seeks an M-RNPs for $(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$. □

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 [25, 26], but admits several polynomial time approximation algorithms [7, 25]. Our approximation algorithms for RNPs rely on solving instances of the 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 vertices $u, v \in V$, 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 vertices $u, v \in V$, H contains at least $c(u, v)$ vertex-disjoint paths between u and v.

Computational Complexity:
Since the RNPc problem studied in Section 3 (which only requires connectivity, rather than biconnectivity) and the *single-tiered fault tolerant relay node placement problem* (1tFTP) studied in [32] and [15] (which can be viewed as the *unconstrained version* of the RNPs problem) are both known to be NP-hard, we believe that the RNPs problem is NP-hard. Instead of deriving a hardness proof of the problem, we concentrate on the design and analysis of polynomial time approximation algorithms that have small approximation ratios.

Our Contributions:
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 $\mathbf{8}$-approximation algorithm for the general RNPs problem, and a $\mathbf{7}$-approximation algorithm for the special RNPs problem where $\mathcal{B} = \emptyset$. Table II lists the most closely related results on this topic.

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
[32]	2	$R \geq r$			14
[32]	2	$R \geq r$	$\surd$		16
this	2	$R \geq r$		$\surd$	**9**
this	2	$R \geq r$	$\surd$	$\surd$	**10**

B. A Framework of Efficient Approximation Algorithms

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

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

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

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: An $\mathsf{F\text{-}RNPs}$ 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 $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ **then**
3: The RNPs 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 vertices in G in the following way. Let u and v be two vertices. 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 $\mathcal{Y}_{\mathcal{A}} = \mathcal{Z} \cap V(\mathcal{H}_{\mathcal{A}})$.

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:

- $\mathsf{RNPs}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ has a feasible solution if and only if $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ has a biconnected component that contains all nodes in $\mathcal{B} \cup \mathcal{X}$.
- When $\mathsf{RNPs}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$ has a feasible solution, Algorithm 2 guarantees computing a feasible solution which uses no more than $\frac{\beta}{2}(\Delta(\mathcal{H}_{opt}) + 4)$ times the number of RNs required in an optimal solution $\mathcal{Y}_{opt}$, where $\mathcal{H}_{opt}$ is a minimum weight biconnected subgraph of $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{opt})$ which spans all nodes in $\mathcal{B} \cup \mathcal{X} \cup \mathcal{Y}_{opt}$, 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 $\mathcal{H}_{opt}$ is a feasible solution to $\{0, 1, 2\}$-SNDP, and $\mathcal{A}$ is a β-approximation algorithm for $\{0, 1, 2\}$-SNDP, we have

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

We need to find an upper bound on $w(\mathcal{H}_{opt})$ using a function of $|\mathcal{Y}_{opt}|$. Let $w_2(\mathcal{H}_{opt})$ denote the total weights of the 2-weight edges in $\mathcal{H}_{opt}$, and let $w_1(\mathcal{H}_{opt})$ denote the total weights of the 1-weight edges in $\mathcal{H}_{opt}$. We have $w(\mathcal{H}_{opt}) = w_2(\mathcal{H}_{opt}) + w_1(\mathcal{H}_{opt})$. Since each RN in $\mathcal{H}_{opt}$ is incident with at most $\Delta(\mathcal{H}_{opt})$ 1-weight edges in $\mathcal{H}_{opt}$, we have

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

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

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

It follows from Lemma 2.1 that

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

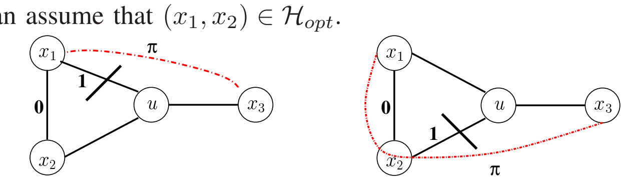
$$\leq \frac{\beta}{2} w(\mathcal{H}_{opt}) \leq \frac{\beta}{2}(4 + \Delta(\mathcal{H}_{opt}))|\mathcal{Y}_{opt}|. \tag{4.5}$$

This proves the theorem. ■

There are several choices for the approximation algorithm $\mathcal{A}$. For example, if we use the algorithm of [7], the corresponding approximation ratio is $\beta = 2$. If we use the algorithm of [25], the corresponding approximation ratio is $\beta = 3$. Next we will find a bound for $\Delta(\mathcal{H}_{opt})$.

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

PROOF. We prove this by contradiction. Assume that RN u is connected to six sensor nodes $x_1, x_2, \ldots, x_6$ in $\mathcal{H}_{opt}$. 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 $\mathsf{HCG}(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{opt})$. Since the weight of (x_1, x_2) is 0, we can assume that $(x_1, x_2) \in \mathcal{H}_{opt}$.



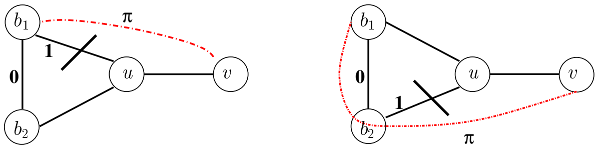
(a) delete edge (x_1, u) (b) delete edge (x_2, u)

Fig. 3. Proof of Lemma 4.1: $\Delta_s(\mathcal{H}_{opt}) \leq 5$

Since $\mathcal{H}_{opt}$ is biconnected, it contains an x_1–x_3 path π which does not go through u. If path π does not go through node x_2 (as shown in Fig. 3(a)), $\mathcal{H}_{opt}$ 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 $\mathcal{H}_{opt}$ will reduce its weight without destroying its biconnectivity [29]. This contradicts the minimum weight property of $\mathcal{H}_{opt}$. If path π goes through node x_2 (as shown in Fig. 3(b)), $\mathcal{H}_{opt}$ 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 $\mathcal{H}_{opt}$ will reduce its weight without destroying its biconnectivity [29]. This again contradicts the minimum weight property of $\mathcal{H}_{opt}$.

Now assume that RN u is connected to two BSs b_1 and b_2 in $\mathcal{H}_{opt}$. Since $\mathcal{Y}_{opt}$ is an optimal solution, u is connected to an SN or another RN v in $\mathcal{H}_{opt}$. Since the weight of (b_1, b_2) is 0, we can assume that $(b_1, b_2) \in \mathcal{H}_{opt}$.

Since $\mathcal{H}_{opt}$ is biconnected, it contains a b_1–v path π which does not go through u. If path π does not go through node b_2



(a) delete edge (b_1, u) (b) delete edge (b_2, u)

Fig. 4. Proof of Lemma 4.1: $\Delta_b(\mathcal{H}_{opt}) \le 1$

(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) which has a weight of 1. Deleting the chord (x_1, u) from $\mathcal{H}_{opt}$ will reduce its weight without destroying its biconnectivity [29]. This contradicts the minimum weight property of $\mathcal{H}_{opt}$. 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 $\mathcal{H}_{opt}$. ■

Corollary 4.1: The general RNPs problem has a **10**-approximation algorithm with a polynomial running time. The special RNPs 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 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 vertex set and edge set of HCG$(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Z})$, and $\alpha(\cdot)$ is the inverse Ackermann function [6]. The special case of RNPs 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 [25, 26]. ■

Note that the **9**-approximation for the *constrained* problem compares favorably with the best-known 10-approximation algorithm for the unconstrained problem [15].

5. Numerical Results

To verify the effectiveness of the frameworks presented in this paper, we implemented Algorithm 1 with $\mathcal{A}$ being the MST based 2-approximation in [18] for STP (simpler than the algorithm in [27]), Algorithm 2 with $\mathcal{A}$ being the sequential maximum flow based 3-approximation in [25] for $\{0, 1, 2\}$-SNDP (simpler than the algorithm in [7]). Our implemented approximations algorithms for RNPc and RNPs are denoted by ARNPc and ARNPs, respectively.

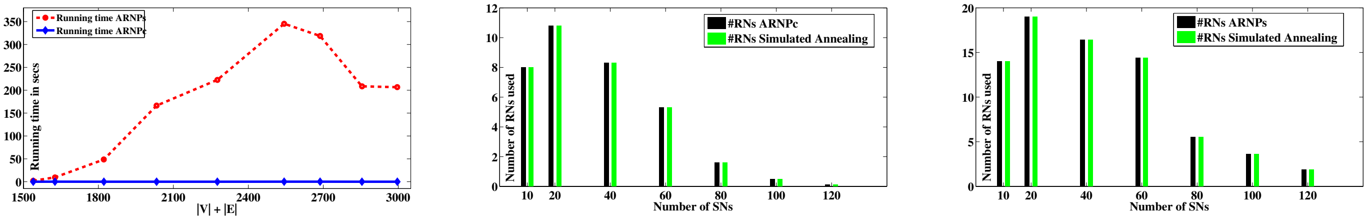
Since there is no previous algorithm for solving these problems, and that optimal solutions are difficult to obtain, we used simulated annealing [17] to obtain *putative* optimal solutions for comparison. For simulated annealing, the initial temperature was set to 100, the number of iterations at a temperature was set to 4000, and the temperature reduction factor was set to 0.8. We kept a bit-vector of length $|\mathcal{Z}|$, where 1 means an RN is placed at the corresponding location and 0 means no RN is placed at the corresponding location. The perturbations were performed by randomly choosing a quarter of the bit-vector and randomly deciding the values of those bits. We used the solutions obtained by our approximation algorithms as the initial solutions for simulated annealing. With these settings, simulated annealing took about 10 times as long as our approximation algorithms. We use SRNPc and SRNPs to denote simulated annealing for RNPc and RNPs, respectively. The tests were run on a 2.4 GHz Linux PC.

As in [15] and [32], the SNs $\mathcal{X}$ were randomly distributed in a square playing field. Two base stations were randomly deployed in the square. We used both regular grid points as the candidate locations for the relay nodes, and randomly generated candidate locations for the relay nodes, and obtained similar results. For brevity, we present the results with regular grid points only. In this setting, 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 set $r = 15$ and $R = 30$.

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 to the area of the field. For the increasing density case, we chose a constant field size of 100×100 sq. units. For the constant density case, we let the size of the playing field increase with the number of SNs.

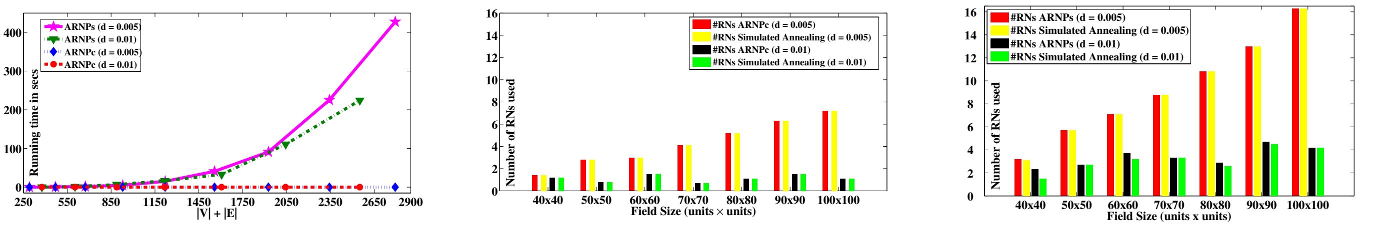
In the case with increasing density, the number of candidate RN locations was 121. The number of SNs was varied from 10 to 130. For each setting the results were averaged over 10 test cases. Fig. 5(a) shows the running time of ARNPs and ARNPc. The X-axis is the sum of the average number of edges and vertices 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 (blue) line shows the running time of ARNPc, which is less than 1 second in all cases. The dashed (red) line shows the running time of ARNPs. The running time 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 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. Figs. 5(b) and 5(c) show the average number of RNs required by ARNPc and ARNPs respectively and that required by SRNPc and SRNPs respectively. Simulated annealing had much longer running times, but only found slightly better solutions in a few cases. This indicates that our approximation algorithms perform well.

For the case of constant density, we studied two sub-cases: one with density $d_1 = 0.005$ and the other with $d_2 = 0.01$. For each density value, we used 7 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. The result of each configuration was averaged over 10 test cases. Fig. 6(a) shows the running times of ARNPc and ARNPs. For both densities, ARNPc has running time less than 1 second. 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



(a) running times of ARNPc and ARNPs (b) number of RNs used by ARNPc, SRNPc (c) number of RNs used by ARNPs, SRNPs

Fig. 5. 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, SRNPc (c) number of RNs used by ARNPs, SRNPs

Fig. 6. Results with constant density: seven different playing fields, from 40×40 to 100×100; two density values, $d_1 = 0.005$ and $d_2 = 0.01$.

density, more pairs of SNs are already biconnected. Hence our algorithm runs faster. Figs. 6(b) and 6(c) show the number of RNs required by various algorithms. Our algorithms perform almost as well as simulated annealing.

6. Conclusions

We have studied the single-tiered constrained relay node placement problem in a hybrid wireless sensor network to meet connectivity and survivability requirements. For each of the two problems, we have 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.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci; Wireless sensor networks: a survey; *COMNET*; Vol. 38(2002), pp. 393–422.

[2] J. Bredin, E. Demaine, M. Hajiaghayi and D. Rus; Deploying sensor networks with guaranteed capacity and fault tolerance; *Mobihoc'05*; 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; *Journal of Global Optimization*; Vol. 18(2000), pp. 17–33.

[4] P. Cheng, C.N. Chuah and X. Liu; Energy-aware node placement in wireless sensor networks; *Globecom'04*; pp. 3210–3214.

[5] X. Cheng, D.Z. Du, L. Wang and B. Xu; Relay sensor placement in wireless sensor networks; *ACM/Springer WINET*; DOI: 10.1007/s11276-006-0724-8.

[6] T. Cormen, C. Leiserson, R. Rivest and C. Stein; *Introduction to Algorithms* (2nd ed); MIT Press and McGraw-Hill, 2001.

[7] L. Fleischer; A 2-Approximation for minimum cost {0, 1, 2} vertex connectivity; *IPCO'01*; pp. 115–129.

[8] M.R. Garey and D.S. Johnson; *Computers and Intractability: A Guide to the Theory of NP-Completeness*; W.H Freeman and Co., 1979.

[9] G. Gupta and M. Younis; Fault tolerant clustering of wireless sensor networks; *WCNC'03*, 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; *Infocom'07*; pp.1667–1675.

[11] B. Hao, J. Tang and G. Xue; Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation; *HPSR'04*; 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; *Secon'05*; pp. 295–304.

[13] F.K. Hwang, D.S. Richards and P. Winter; *The Steiner Tree Problem*; Annals of Discrete Mathematics, 1992

[14] B. Karp and H. Kung; GPSR: greedy perimeter stateless routing for wireless networks; *Mobicom'00*, pp. 243–254.

[15] A. Kashyap, S. Khuller and M. Shayman; Relay placement for higher order connectivity in wireless sensor networks; *Infocom'06*.

[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 Transactions on Computers*; Vol. 56(2007), pp. 710–715.

[17] S. Kirkpatrick, C.D. Gelatt, Jr. and M.P. Vecchi; Optimization by simulated annealing; *Science*; Vol. 220(1983), pp. 671–680.

[18] L.T. Kou, G. Markowsky and L. Berman; A fast algorithm for Steiner trees; *Acta Informatica*; Vol. 15(1981); pp. 141–145.

[19] Q. Li, J. Aslam, D. Rus; Online power-aware routing in wireless ad-hoc networks; *Mobicom'01*, pp. 97–107.

[20] G. Lin and G. Xue; Steiner tree problem with minimum number of Steiner points and bounded edge-length; *Information Processing Letters*; Vol. 69(1999), pp. 53-57.

[21] H. Liu, P.J. Wan and X.H. Jia; Fault-tolerant relay node placement in wireless sensor networks; *LNCS*; Vol. 3595(2005), pp. 230–239.

[22] E. Lloyd and G. Xue; Relay node placement in wireless sensor networks; *IEEE Transactions on Computers*; Vol. 56(2007), pp. 134–138.

[23] W. Lou, W. Liu and Y. Fang; SPREAD: enhancing data confidentiality in mobile ad hoc networks; *Infocom'04*; pp. 2404–2413.

[24] J. Pan, Y.T. Hou, L. Cai, Y. Shi, S.X. Shen; Topology control for wireless sensor networks; *Mobicom'03*, pp. 286–299.

[25] R. Ravi and D.P. Williamson; An approximation algorithm for minimum-cost vertex-connectivity problems; *Algorithmica*;Vol.18(1997),pp.21-43.

[26] R. Ravi and D.P. Williamson; Erratum: an approximation algorithm for minimum-cost vertex-connectivity problems; *Algorithmica*; Vol. 34(2002), pp.98–107.

[27] G. Robins and A. Zelikovsky; Tighter bounds for graph Steiner tree approximation; *SIAM J. on Disc. Math.*; Vol. 19(2005), pp. 122–134.

[28] A. Srinivas, G. Zussman, and E. Modiano; Mobile backbone networks–Construction and maintenance, *Mobihoc'06*, pp. 166–177.

[29] D.B. West, *Introduction to Graph Theory*, Prentice Hall, 1996.

[30] K. Xu, H. Hassanein, G. Takahara and Q. Wang; Relay node deployment strategies in heterogeneous wireless sensor networks: multiple-hop communication case; *Secon'05*, pp. 575-585.

[31] O.Younis and S.Fahmy; Distributed clustering for ad-hoc sensor networks: a hybrid, energy-efficient approach; *Infocom'04*; pp.269–640.

[32] W. Zhang, G. Xue, and S. Misra; Fault-tolerant relay node placement in wireless sensor networks: problems and algorithms; *Infocom'07*; pp.1649–1657.