Fault-Tolerant Relay Node Placement in Wireless Sensor Networks: Problems and Algorithms

Weiyi Zhang, Guoliang Xue and Satyajayant Misra

***Abstract*— Two fundamental functions of the sensor nodes in a wireless sensor network are to sense its environment and to transmit sensed information to a basestation. One approach to prolong sensor network lifetime is to deploy some *relay nodes* whose main function is to communicate with the sensor nodes, other relay nodes, and the basestations. It is desirable to deploy a minimum number of relay nodes to achieve certain connectivity requirement. In this paper, we study four related fault-tolerant relay node placement problems, each of which has been previously studied only in some restricted form. For each of them, we discuss its computational complexity and present a polynomial time $O(1)$-approximation algorithm with a small approximation ratio. When the problem reduces to a previously studied form, our algorithm either improves the previous best algorithm or reduces to the previous best algorithm.**

Keywords: **Survivable relay placement, wireless sensor networks.**

1. Introduction and Related Work

A wireless sensor network (WSN) consists of many low-cost and low-power *sensor nodes* (SNs)[1]. There has been extensive research on energy aware routing [4, 10, 13, 16, 27], improvement in lifetime [12, 21, 24, 26], and survivability [20, 22]. 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 costly, but more powerful *relay nodes* (RNs) whose main function is to communicate with the SNs and other RNs [2, 5, 9, 12, 14, 18, 19, 23, 26]. This is the subject of study of this paper.

We first review prior works on *single-tiered relay node placement*, where both SNs and RNs participate in packet forwarding. Cheng *et al.* [5] proposed to deploy a minimum number of RNs in a WSN so that between every pair of SNs, there is a path consisting of RNs and/or SNs where each hop of the path is no longer than the common transmission range $r > 0$ of the SNs. This problem is equivalent to the *Steiner minimum tree with minimum number of Steiner points and bounded edge length* problem (SMT-MSPBEL), defined by Lin and Xue in the study of amplifier placement in optical networks [17], where they proved the problem is NP-hard and presented a minimum spanning tree (MST) based 5-approximation algorithm. In [3], Chen *et al.* proved that the Lin-Xue algorithm is a 4-approximation algorithm. They also presented a 3-approximation algorithm. In [5], Cheng *et al.* presented a faster 3-approximation algorithm. In [2], Bredin *et al.* extended the relay node placement problem studied in [3, 5, 17] to the case of k-connectivity, instead of 1-connectivity, and presented polynomial time $O(1)$-approximation algorithms for any fixed k. In [14], Kashyap *et al.* presented a 10-approximation algorithm ensuring 2-connectivity. All of the above works assume that the transmission range of the RNs is the same as that of the SNs. In [19], Lloyd and Xue generalized the problem studied in [3, 5, 17] to the case where the RNs have transmission range $R \geq r$, and presented a 7-approximation algorithm.

Next we review prior works on *two-tiered relay node placement*, where only the RNs participate in packet forwarding. Motivated by the works [8] and [21] on two-tiered WSNs, Hao *et al.* in [9] formulated 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 [23] 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 [18], under the assumption that $R = r$, but 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 [19], 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.

In this paper, we study both single-tiered and two-tiered relay node placement problems that ensure 2-connectivity, under the mild condition $R \geq r$, and no assumption on the distribution of the SNs. For the single-tiered problem (which contains the problem studied in [14] as a special case), we present a 14-approximation algorithm. Our algorithm reduces to that of [14] when the problem is reduced to the problem studied in [14]. For the two-tiered problem, we present a $(10+\epsilon)$-approximation algorithm, improving the previous-best $(24 + \epsilon)$-approximation algorithm [18] designed for a special case ($R = r$). We then generalize the two relay node placement problems to cases where there are also some basestations (BSs), and present a 16-approximation algorithm for the single-tiered problem with BSs and a $(20 + \epsilon)$-approximation algorithm for the two-tiered problem with BSs.

In Section 2, we present basic notations. In Section 3, we study single-tiered fault-tolerant relay node placement problems. In Section 4, we study two-tiered fault-tolerant relay node placement problems. We present numerical results in Section 5 and conclude the paper in Section 6.

2. Notations and Basic Concepts

We use BS, SN, RN to denote *basestation*, *sensor node*, and *relay node*, respectively. For two points x and y in the plane, we use $d(x, y)$ to denote the Euclidean distance between them, and use $[x, y]$ to denote the line segment connecting them. We use $\mathcal{Y} = \{y_1, \ldots, y_l\}$ to denote $l \geq 0$ RNs, $\mathcal{X} = \{x_1, \ldots, x_n\}$

This research was supported in part by ARO grant W911NF-04-1-0385 and NSF grants CCF-0431167 and ANI-0312635. The information reported here does not reflect the position or the policy of the federal government.

Weiyi Zhang, Guoliang Xue and Satyajayant Misra are all with the Department of Computer Science and Engineering at Arizona State University, Tempe, AZ 85287. Email: {weiyi.zhang, xue, satyajayant}@asu.edu.

to denote $n \geq 1$ SNs, and $\mathcal{B} = \{b_1, \ldots, b_m\}$ to denote $m \geq 0$ BSs. We use the same symbol to denote the BS/SN/RN and its corresponding position in the plane. For example, $d(x_i, y_j)$ is the Euclidean distance between SN $x_i \in \mathcal{X}$ and RN $y_j \in \mathcal{Y}$, $[x_i, x_j]$ is the line segment connecting SNs x_i and x_j.

For graph theoretic terms not defined in this paper, we refer readers to the standard textbook [25]. 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, 7].

A polynomial time α-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 α 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* α. $\mathcal{A}_\epsilon$ is a *polynomial time approximation scheme* (PTAS) for a minimization problem, if for any fixed $\epsilon > 0$, $\mathcal{A}_\epsilon$ is a polynomial time $(1+\epsilon)$-approximation algorithm with ϵ treated as a constant. Since the running time of our algorithms may also depend on the size of the output (e.g. the number of RNs to be deployed), we say an algorithm has polynomial running time if the running time is bounded by a polynomial in the input size *and the output size* of the instance. The acronym WLOG stands for "*without loss of generality*".

3. Single-Tiered Fault-Tolerant Relay Placement

A. Problem Definitions and Summary of Results

Single-Tiered Placement with Basestations:

Definition 3.1: 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 for 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 edge-weighted 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 $(b_i, b_j) = (b_j, b_i)$, with *length* $l(b_i, b_j) = 0$.
- 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 $d(y, z) \leq R$. The length of edge $(y, z) \in E$ is $l(y, z) = d(y, z)$.
- For an SN $x \in \mathcal{X}$ and a node $z \in \mathcal{B} \cup \mathcal{X} \cup \mathcal{Y}$ which could be either an SN, an RN or a BS, E contains the undirected edge $(x, z) = (z, x)$ if and only if $d(x, z) \leq r$. The length $l(x, z)$ of edge $(x, z) \in E$ is 0 if z is an SN and $d(x, z)$ otherwise.

An edge connecting two SNs is called an SN-SN edge. We can similarly define SN-RN, SN-BS, RN-RN, RN-BS, and BS-BS edges. The edge length function naturally generalizes to the *length of a subgraph* of HCG by summation. Note that our definition of *length* facilitates the proof of Lemma 3.3. □

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, and $\mathcal{X}$ be a set of SNs. A set of RNs $\mathcal{Y} = \{y_1, \ldots, y_l\}$ is said to be a *feasible single-tiered fault-tolerant relay node placement with basestations* (denoted by F1tFTPB) for $(r, R, \mathcal{B}, \mathcal{X})$ if the graph $HCG(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y})$ is 2-connected. The *size* of the corresponding F1tFTPB is $|\mathcal{Y}|$. An F1tFTPB is said to be a *minimum single-tiered fault-tolerant relay node placement with basestations* for $(r, R, \mathcal{B}, \mathcal{X})$ (denoted by M1tFTPB) if it has the minimum size among all F1tFTPBs for $(r, R, \mathcal{B}, \mathcal{X})$. The *single-tiered fault-tolerant relay node placement problem with basestations* for $(r, R, \mathcal{B}, \mathcal{X})$, denoted by 1tFTPB$(r, R, \mathcal{B}, \mathcal{X})$, seeks an M1tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$.□

Single-Tiered Placement without Basestations:
We also study a special case of the 1tFTPB problem where $\mathcal{B} = \emptyset$. In this case, the graph $HCG(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y})$ defined above becomes $HCG(r, R, \mathcal{X}, \mathcal{Y})$. Similarly,

- the term *feasible single-tiered fault-tolerant relay node placement with basestations* (F1tFTPB) for $(r, R, \mathcal{B}, \mathcal{X})$ becomes *feasible single-tiered fault-tolerant relay node placement* (F1tFTP) for $(r, R, \mathcal{X})$;
- the term *minimum single-tiered fault-tolerant relay node placement with basestations* (M1tFTPB) for $(r, R, \mathcal{B}, \mathcal{X})$ becomes *minimum single-tiered fault-tolerant relay node placement* (M1tFTP) for $(r, R, \mathcal{X})$;
- the term *single-tiered fault-tolerant relay node placement with basestations problem* (1tFTPB) for $(r, R, \mathcal{B}, \mathcal{X})$ becomes *single-tiered fault-tolerant relay node placement problem* (1tFTP) for $(r, R, \mathcal{X})$.

Discussions:
To our knowledge, neither the 1tFTPB$(r, R, \mathcal{B}, \mathcal{X})$ problem nor its special case 1tFTP$(r, R, \mathcal{X})$ has been studied before, although a restricted version of the problems, 1tFTP$(r, r, \mathcal{X})$ (with $R = r$), has been well studied in the literature [2, 14]. Since the 1tFTP$(r, r, \mathcal{X})$ problem is NP-hard [14], both 1tFTP$(r, R, \mathcal{X})$ and 1tFTPB$(r, R, \mathcal{B}, \mathcal{X})$ are NP-hard.

In a Mobihoc'2005 paper [2], Bredin *et al.* presented an $O(1)$-approximation algorithm for the problem of deploying a minimum number of RNs in a WSN to ensure k-connectivity, for any constant k, under the assumption that $R = r$. Their algorithm uses an α-approximation algorithm for computing a minimum weight k-connected spanning subgraph, and has an approximation ratio bounded by $(9k^4 + 36(k^3 + k^2))\alpha$. This bound is highly dependent on the geometric properties implied by the restriction $R = r$, and cannot be easily extended to the general case of $R \geq r$. For $k = 2$, it is known that $\alpha = 2$ [15].

In an Infocom'2006 paper [14], Kashyap *et al.* presented a 10-approximation algorithm for 1tFTP$(r, r, \mathcal{X})$. Again, the bound on the approximation ratio is highly dependent on the geometric properties implied by the restriction $R = r$.

Since RNs generally have more energy and stronger communication power than the SNs, the 1tFTP$(r, R, \mathcal{X})$ problem is a more realistic model than the 1tFTP$(r, r, \mathcal{X})$ problem that has been well studied.

We also study the more general 1tFTPB$(r, R, \mathcal{B}, \mathcal{X})$ problem because a WSN is usually connected to one or more BSs and that BSs are more powerful than the SNs and the RNs.

Results:
We will present a simple 14-approximation algorithm for the 1tFTP$(r, R, \mathcal{X})$ problem, and a simple 16-approximation algorithm for the 1tFTPB$(r, R, \mathcal{B}, \mathcal{X})$ problem.

B. *Approximation Algorithm for* 1tFTP$(r, R, \mathcal{X})$

All of our algorithms are based on a concept known as *steinerization*, which was first introduced by Lin and Xue [17] for the case of $R = r$ and later generalized by Lloyd and Xue [19] for the case of $R \geq r$.

Let x_i and x_j be two SNs. If $d(x_i, x_j) \leq r$, x_i and x_j can communicate with each other directly. If $d(x_i, x_j) > r$, we can *connect* x_i and x_j by deploying a minimum number of RNs on the line segment $[x_i, x_j]$ (*steinerizing* $[x_i, x_j]$) in the following way:

- If $d(x_i, x_j) \in (r, 2r]$, place one RN at the midpoint of line segment $[x_i, x_j]$.
- If $d(x_i, x_j) > 2r$, place $1 + \lceil \frac{d(x_i,x_j)-2r}{R} \rceil$ RNs on the line segment $[x_i, x_j]$ such that one RN (call it y_F) is at distance r from x_i, one RN (call it y_L) is at distance r from x_j, and the other $\lceil \frac{d(x_i,x_j)-2r}{R} \rceil - 1$ RNs evenly divide the line segment $[y_F, y_L]$ into $c(x_i, x_j) - 1$ segments with length bounded by R.

Definition 3.3: Given communication ranges $R \geq r > 0$ and a set of SNs $\mathcal{X}$, the 3-tuple $(r, R, \mathcal{X})$ induces an edge weighted undirected complete graph $G^S(r, R, \mathcal{X})$, called the *steinerized graph of* $(r, R, \mathcal{X})$, with vertex set $V = \mathcal{X}$ and edge cost defined in the following.

$$c(x_i, x_j) = \begin{cases} 0, & \text{if } d(x_i, x_j) \in [0, r]; \\ 1, & \text{if } d(x_i, x_j) \in (r, 2r]; \\ 1 + \lceil \frac{d(x_i,x_j)-2r}{R} \rceil, & \text{otherwise.} \end{cases} \tag{3.1}$$

Essentially, $c(x_i, x_j)$ is the number of RNs needed to connect SNs x_i and x_j by steinerizing $[x_i, x_j]$. The edge cost function generalizes naturally to the *cost of a subgraph* of $G^S(r, R, \mathcal{X})$ by summation. □

Our approximation algorithm for 1tFTP$(r, R, \mathcal{X})$ consists of three main steps. First, we construct the steinerized graph $G^S(r, R, \mathcal{X})$. Then, we compute G_A, a 2-approximation to a minimum cost 2-connected spanning subgraph of $G^S(r, R, \mathcal{X})$. Finally, we deploy the relay nodes by steinerizing each of the edges of G_A.

Algorithm 1 Approximation Algorithm for 1tFTP$(r, R, \mathcal{X})$

Input: $R \geq r > 0$ and SNs $\mathcal{X} = \{x_1, \ldots, x_n\}$.
Output: Relay nodes $\mathcal{Y}_A = \{y_1, \ldots, y_l\}$.

```
Construct the steinerized graph G^S(r, R, X).
Compute a 2-connected spanning subgraph G_A of G^S(r, R, X) using the algorithm A of [15].
l := 0;
for each edge (x_i, x_j) ∈ G_A s.t. c(x_i, x_j) ≥ 1 do
    Steinerize edge (x_i, x_j) with c(x_i, x_j) relay nodes: y_{l+1}, y_{l+2}, ..., y_{l+c(x_i,x_j)};
    l := l + c(x_i, x_j).
end for
```

Theorem 3.1: Algorithm 1 is a 14-approximation algorithm for 1tFTP. It can compute a 2-connected spanning subgraph G_A of $G^S(r, R, \mathcal{X})$ such that $c(G_A) \leq 14 \cdot |\mathcal{Y}_{opt}|$ in $O(n^4)$ time, and requires additional $O(n^2 + |\mathcal{Y}_{opt}|)$ time to deploy the RNs in $\mathcal{Y}_A$, where $\mathcal{Y}_{opt}$ is an M1tFTP for $(r, R, \mathcal{X})$. □
We need a sequence of lemmas before proving this theorem.

Lemma 3.1: The RNs $\mathcal{Y}_A$ placed by Algorithm 1 is an F1tFTP for $(r, R, \mathcal{X})$. Let $\mathcal{Y}_B$ be any F1tFTP for $(r, R, \mathcal{X})$ that is obtained by steinerizing the edges of a 2-connected spanning subgraph of $G^S(r, R, \mathcal{X})$. Then $|\mathcal{Y}_A| \leq 2 \cdot |\mathcal{Y}_B|$. □
PROOF. G_A is a 2-connected spanning subgraph which spans all the SNs $\mathcal{X}$. The steinerization of an edge (x_i, x_j) with $c(x_i, x_j)$ RNs can be viewed as a sequence of $c(x_i, x_j)$ edge *subdivision* operations [25]. Therefore the resulting hybrid communication graph $HCG(r, R, \mathcal{X}, \mathcal{Y}_A)$ is 2-connected. This shows that $\mathcal{Y}_A$ is an F1tFTP for $(r, R, \mathcal{X})$.

Let G_{min} be a minimum cost 2-connected spanning subgraph of $G^S(r, R, \mathcal{X})$. Then its cost is $c(G_{min}) \leq |\mathcal{Y}_B|$, since the number of RNs needed to steinerize edge (x_i, x_j) is $c(x_i, x_j)$. Since $\mathcal{A}$ is a 2-approximation algorithm, we have $c(G_A) \leq 2 \cdot c(G_{min}) \leq 2 \cdot |\mathcal{Y}_B|$. ■

Definition 3.4: Let $\mathcal{Y}$ be an F1tFTP for $(r, R, \mathcal{X})$ and $\mathcal{L}(\mathcal{Y})$ be a 2-connected spanning subgraph of $HCG(r, R, \mathcal{X}, \mathcal{Y})$. We call $\mathcal{L}(\mathcal{Y})$ a *layout* of the F1tFTP $\mathcal{Y}$. Note that the *length* of $\mathcal{L}(\mathcal{Y})$ is $l(\mathcal{L}(\mathcal{Y}))$ (see Definition 3.1). □

Definition 3.5: A layout $\mathcal{L}(\mathcal{Y})$ is called a *shortest layout for single-tiered fault-tolerant relay node placement* for $(r, R, \mathcal{X})$, denoted by S1tFTL, if $\mathcal{Y}$ is an M1tFTP for $(r, R, \mathcal{X})$ and $\mathcal{L}(\mathcal{Y})$ has the minimum length among all layouts of M1tFTPs for $(r, R, \mathcal{X})$. □

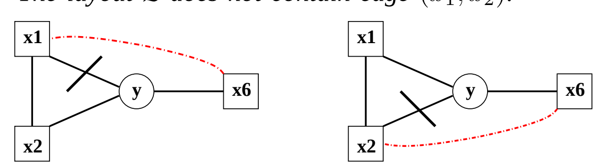
Definition 3.6: Let $\mathcal{L} = \mathcal{L}(\mathcal{Y})$ be a layout of an F1tFTP for $(r, R, \mathcal{X})$. Let $y \in \mathcal{Y}$ be an RN. The *sensor degree* of y in the layout $\mathcal{L}(\mathcal{Y})$, denoted by $\delta_s(y, \mathcal{L})$, is the number of SNs in $\mathcal{X}$ that are adjacent with y in $\mathcal{L}(\mathcal{Y})$. The *relay degree* of y in the layout $\mathcal{L}(\mathcal{Y})$, denoted by $\delta_r(y, \mathcal{L})$, is the number of RNs in $\mathcal{Y}$ that are adjacent with y in $\mathcal{L}(\mathcal{Y})$. □

Lemma 3.2: Let $\mathcal{L} = \mathcal{L}(\mathcal{Y})$ be an S1tFTL for $(r, R, \mathcal{X})$. Then $\delta_s(y, \mathcal{L}) \leq 5$ for any RN $y \in \mathcal{Y}$. □
PROOF. Note that by our assumption, $\mathcal{Y}$ is an M1tFTP and that $\mathcal{L}$ has the shortest length among all layouts of M1tFTPs. Assume that there is an RN $y \in \mathcal{Y}$ such that $\delta_s(y, \mathcal{L}) \geq 6$. We will show that this assumption leads to the existence of another layout $\mathcal{L}'$ of $\mathcal{Y}$ such that $l(\mathcal{L}') < l(\mathcal{Y})$, contradicting the *shortest length* assumption of $\mathcal{L}$.

Since $\delta_s(y, \mathcal{L}) \geq 6$, there exist at least six SNs that are adjacent with y in $\mathcal{L}$. WLOG, assume that $x_1, x_2, \ldots, x_6$ are six SNs that are adjacent with y in $\mathcal{L}$ and that $\angle x_1 y x_2 \leq 60^o$.

We first prove the following proposition.

(a): *The layout $\mathcal{L}$ does not contain edge (x_1, x_2).*



(a) (y, x_1) can be cut (b) (y, x_2) can be cut

Fig. 1. $\mathcal{L}$ cannot contain edge (x_1, x_2).

Since $\mathcal{L}$ is 2-connected, there is a path π in $\mathcal{L}$ connecting x_6 and x_1 without using node y. If π does not go through x_2, we have a scenario as shown in Fig. 1(a). If π goes through x_2, we have a path π' in $\mathcal{L}$ connecting x_6 and x_2 without using nodes y and x_1, as shown in Fig. 1(b). In the first scenario (see Fig. 1(a)), $\mathcal{L}$ contains a cycle going through x_1, x_2, y, and x_6 and a chord (y, x_1). Deleting the chord (y, x_1) from $\mathcal{L}$ will reduce the length without destroying 2-connectivity [25], contradicting the shortest length assumption of $\mathcal{L}$. Similarly, deleting the chord (y, x_2) will lead to a contradiction in the second scenario (refer to Fig. 1(b)). This proves (a).

We need to prove the following claim.

(b): *By replacing (y, x_1) with (x_1, x_2) in $\mathcal{L}$, we can obtain another 2-connected spanning subgraph $\mathcal{L}'$ of $HCG(r, R, \mathcal{X}, \mathcal{Y})$, with $l(\mathcal{L}') < l(\mathcal{L})$.*

It follows from (a) that $\mathcal{L}$ does not contain edge (x_1, x_2). Let $\mathcal{L}'$ be obtained from $\mathcal{L}$ by replacing edge (y, x_1) with edge (x_1, x_2). Since $d(y, x_1) \leq r$, $d(y, x_2) \leq r$ and $\angle x_1 y x_2 \leq 60^o$, we have $d(x_1, x_2) \leq r$. Since $l(y, x_1) = d(y, x_1)$ and $l(x_1, x_2) = 0$, we have $l(\mathcal{L}') = l(\mathcal{L}) - d(y, x_1) < l(\mathcal{L})$. We have to show that $\mathcal{L}'$ is also a 2-connected spanning subgraph of $HCG(r, R, \mathcal{X}, \mathcal{Y})$.

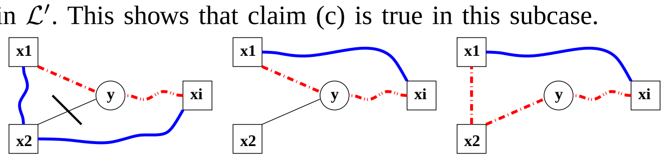
We first claim that,

(c): *for each pair of* SN*s $x_i, x_j \in \mathcal{X}$, there exists a pair of node disjoint paths π_1, π_2 in $\mathcal{L}'$ connecting x_i and x_j.*

Since $\mathcal{L}$ is a 2-connected spanning subgraph of $HCG(r, R, \mathcal{X}, \mathcal{Y})$, there exists a pair of node disjoint paths π_1 and π_2 in $\mathcal{L}$ connecting x_i and x_j. If neither path uses edge (y, x_1), π_1 and π_2 also form a pair of node disjoint paths in $\mathcal{L}'$. Now we consider the case where one of the paths (WLOG, assuming π_1) uses edge (y, x_1).

First, consider the subcase where $\{x_i, x_j\} = \{x_1, x_2\}$. In this case, π_2 and the edge (x_1, x_2) form two node disjoint x_1–x_2 paths in $\mathcal{L}'$. This shows that (c) is true in this subcase.

Next, consider the subcase where $x_j = x_1$ but $x_i \neq x_2$. Since π_1 goes through y (which is an RN), π_2 does not go through y. If π_2 goes through x_2, $\mathcal{L}$ contains the cycle formed by the two paths π_1 and π_2, as well as a chord (y, x_2). This contradicts the shortest length assumption of $\mathcal{L}$ (see Fig. 2(a) and similar argument used in the proof of (a)). Therefore π_2 does not go through x_2 (see Fig. 2(b)). We can replace π_1 with a new x_i–x_1 path π_3 which goes from x_i to y along π_1, then to x_2 via edge (y, x_2), then to x_1 via edge (x_2, x_1) (see Fig. 2(c)). π_2 and π_3 form a pair of node disjoint x_i–x_1 paths in $\mathcal{L}'$. This shows that claim (c) is true in this subcase.



(a) impossible (b) before (c) after

Fig. 2. Replacing edge (y, x_1) with edge (x_1, x_2)

Using an argument similar to the one used in the above paragraph (also see Fig. 2(a)), we can prove that it is impossible to have $x_j = x_2$ and $x_i \neq x_1$, as it contradicts the shortest length assumption of $\mathcal{L}$.

Finally we consider the subcase where $\{x_i, x_j\} \cap \{x_1, x_2\} = \emptyset$. Since π_1 goes through y, π_2 does not go through y. If π_2 goes through x_2, then $\mathcal{L}$ contains the cycle formed by the two paths π_1 and π_2, as well as a chord (y, x_2), contradicting the shortest length assumption of $\mathcal{L}$. Therefore π_2 does not go through x_2. We can replace π_1 with a new x_i–x_j path π_3 which goes from x_i to y along π_1, then to x_2 via edge (y, x_2), then to x_1 via edge (x_2, x_1), then to x_j following the subpath on π_1. π_2 and π_3 form a pair of node disjoint x_i–x_j paths in $\mathcal{L}'$. This shows that claim (c) is true in this subcase. This completes the proof for claim (c).

Now, we have proved that for any pair of SNs x_i and x_j, there is a pair of node disjoint paths in $\mathcal{L}'$ connecting them. We will show that $\mathcal{L}'$ is actually a 2-connected spanning subgraph of $HCG(r, R, \mathcal{X}, \mathcal{Y})$.

Following (c), for every pair of SNs x_i, x_j, there is a cycle in $\mathcal{L}$ passing through x_i and x_j. Therefore all SNs are on a common biconnected component of $\mathcal{L}'$ [25]. We claim that,

(d): *all* RN*s in $\mathcal{Y}$ are also on the same biconnected component with the* SN*s.*

We prove this by the following simple *coloring scheme*. Initially all RNs in $\mathcal{Y}$ are colored *white*. We examine the $\frac{n(n-1)}{2}$ SN pairs in a given order (e.g. alphabetical order). Whenever we examine a pair of SNs x_i and x_j, we compute a pair of node disjoint paths in $\mathcal{L}$ connecting x_i and x_j. This pair of paths form a cycle which contains x_i and x_j, as well as some RNs in $\mathcal{Y}$. We color all the RNs on this cycle *black*, as they must be on the same biconnected component with x_i and x_j, as well as the rest of the SNs. We claim that all RNs in $\mathcal{Y}$ will be colored *black* at the end of the above coloring scheme, as otherwise, the subset of *black* RNs will also be an F1tFTP for $(r, R, \mathcal{X})$, contradicting the minimum size assumption of $\mathcal{Y}$. Therefore we have proved claim (d). This also completes the proof of claim (b), as well as the lemma. ■

Definition 3.7: Let $\mathcal{L} = \mathcal{L}(\mathcal{Y})$ be a layout of an F1tFTP $\mathcal{Y}$ for $(r, R, \mathcal{X})$. A *Steiner component* of $\mathcal{L}$ is a maximal subgraph $\mathcal{C}$ of $\mathcal{L}$ with the property that for any two nodes u and v in $\mathcal{C}$, there exists a u–v path in $\mathcal{C}$ whose internal nodes are all RNs. □

Lemma 3.3: There exists a 2-connected spanning subgraph G_{appx} of the steinerized graph $G^S(r, R, \mathcal{X})$ such that $c(G_{appx})$, the cost of G_{appx}, is at most 7 times the size of an M1tFTP for $(r, R, \mathcal{X})$. □

PROOF. Let $\mathcal{L}_{opt} = \mathcal{L}(\mathcal{Y}_{opt})$ be an S1tFTL for $(r, R, \mathcal{X})$, where $\mathcal{Y}_{opt}$ is an M1tFTP for $(r, R, \mathcal{X})$. Let $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_l$ be the Steiner components of $\mathcal{L}_{opt}$. Let $|\mathcal{C}_i|$ denote the number of RNs in $\mathcal{C}_i$, $i = 1, 2, \ldots, l$.

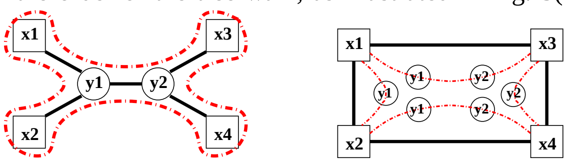
For each component $\mathcal{C}_i$, $i = 1, \ldots, l$, construct a connected (2-connected, in some cases) subgraph G_i of $G^S(r, R, \mathcal{X})$, spanning all the SNs in $\mathcal{C}_i$, in the following way.

Compute a spanning tree $\mathcal{T}_i$ of $\mathcal{C}_i$ such that every leaf node of $\mathcal{T}_i$ is an SN, and every internal node of $\mathcal{T}_i$ is an RN. The existence of such a spanning tree follows from the definition of Steiner components. $\mathcal{T}_i$ can be constructed in the following way: Initialize $\mathcal{T}_i$ to contain any chosen SN x_i^{root} in $\mathcal{C}_i$ and no edge. Let x_i^{new} be any other SN in $\mathcal{C}_i$ which is not added to $\mathcal{T}_i$ yet. There is an x_i^{new}–x^{root} path π^{new} such that all internal nodes of π^{new} are RNs. Let z^{new} be the node on $\mathcal{T}_i$ that is first met if we traverse the path π^{new} from node x^{new}. Grow the tree $\mathcal{T}_i$ by adding the x^{new}–z^{new} subpath of π^{new}.

If $|\mathcal{C}_i| = 0$, $\mathcal{T}_i$ consists of a single edge connecting two SNs x_i^1, x_i^2 such that $d(x_i^1, x_i^2) \leq r$. In this case, we set $G_i = \mathcal{T}_i$. Note that G_i is a subgraph of $G^S(r, R, \mathcal{X})$ s.t. $c(G_i) = |\mathcal{C}_i|$.

If $|\mathcal{C}_i| \geq 1$, but all RNs in $\mathcal{T}_i$ have degree (sensor degree plus relay degree) 2, $\mathcal{T}_i$ must be a path connecting two SNs x_i^1, x_i^2 using $|\mathcal{C}_i|$ internal RNs. Therefore we must have $d(x_i^1, x_i^2) \leq 2r + R \cdot (|\mathcal{C}_i| - 1)$. In this case, we set G_i to contain nodes x_i^1 and x_i^2, and a single edge (x_i^1, x_i^2). Note that G_i is a subgraph of $G^S(r, R, \mathcal{X})$ s.t. $c(G_i) \leq |\mathcal{C}_i|$.

Now, assume that $\mathcal{T}_i$ contains at least one relay node with degree 3 or more. Starting from a sensor node in $\mathcal{T}_i$ and taking a clockwise walk of the tree, we obtain an Eulerian loop, as illustrated in Fig. 3(a). The Eulerian loop induces a ring subgraph G_i (of $G^S(r, R, \mathcal{X})$) connecting the sensor nodes in $\mathcal{T}_i$ in the order of the tree walk, as illustrated in Fig. 3(b).



(a) $\mathcal{T}_i$ and a loop (b) the resulting ring

Fig. 3. Constructing a ring spanning all sensor nodes in component $\mathcal{C}_i$ by taking an Eulerian loop of a spanning tree $\mathcal{T}_i$ of $\mathcal{C}_i$. Note that the number of copies of each relay node is equal to its degree in $\mathcal{T}_i$.

Note that each RN y of $\mathcal{T}_i$ is used exactly $\delta_s(y, \mathcal{T}_i) + \delta_r(y, \mathcal{T}_i)$ times by the Eulerian loop, where $\delta_s(y, \mathcal{T}_i)$ and $\delta_r(y, \mathcal{T}_i)$ are the sensor degree and relay degree of node y in $\mathcal{T}_i$ (see Fig. 3). Following Lemma 3.2, we have $\delta_s(y, \mathcal{T}_i) \leq \delta_s(y, \mathcal{L}) \leq 5$ for each RN y in $\mathcal{T}_i$. Since $\mathcal{T}_i$ is a *tree* with $|\mathcal{C}_i|$ internal (relay) nodes, it contains $|\mathcal{C}_i| - 1$ RN–RN edges. This leads to $\sum_{y \in \mathcal{C}_i} \delta_r(y, \mathcal{T}_i) = 2 \cdot (|\mathcal{C}_i| - 1)$. Therefore the total number of relay nodes needed to steinerize the subgraph G_i is bounded by the following formula.

$$c(G_i) \leq \sum_{y \in \mathcal{C}_i} (\delta_s(y, \mathcal{T}_i) + \delta_r(y, \mathcal{T}_i)) < 5|\mathcal{C}_i| + 2|\mathcal{C}_i| = 7|\mathcal{C}_i|. \quad (3.2)$$

We define G_{appx} as the *union* of $G_1, G_2, \ldots, G_l$ constructed above. Note that G_i is a spanning subgraph of all sensor nodes in $\mathcal{C}_i$ for $i = 1, \ldots, l$, and is a ring (therefore 2-connected) unless $\mathcal{C}_i$ is a path. Therefore the 2-connectivity of $\mathcal{L}$ implies the 2-connectivity of G_{appx}. At the same time, we have (using inequality (3.2))

$$c(G_{appx}) = \sum_{i=1}^{l} c(G_i) \leq \sum_{i=1}^{l} 7|\mathcal{C}_i| = 7|\mathcal{Y}_{opt}|. \quad (3.3)$$

This completes the proof of Lemma 3.3. ■

Now we are ready to prove Theorem 3.1. It follows from Lemma 3.3 that there exists a 2-connected spanning subgraph G_{appx} of $G^S(r, R, \mathcal{X})$ such that $c(G_{appx}) \leq 7 \cdot |\mathcal{Y}_{opt}|$, where $\mathcal{Y}_{opt}$ is an M1tFTP for $(r, R, \mathcal{X})$. It follows from Lemma 3.1 that the F1tFTP $\mathcal{Y}_A$ computed by Algorithm 1 has size

$$|\mathcal{Y}_A| \leq 2 \cdot c(G_{appx}) \leq 14 \cdot |\mathcal{Y}_{opt}|. \quad (3.4)$$

This proves the approximation ratio of Algorithm 1.

Line 1 of the algorithm requires $O(n^2)$ time to construct the complete graph. Line 2 of the algorithm requires $O(n^4)$ time by the 2-approximation algorithm of [15]. Lines 3–7 requires $O(n^2 + |\mathcal{Y}_{opt}|)$ time, where the $O(n^2)$ term is due to the loop over the $O(n^2)$ edges in G_A, and the $O(|\mathcal{Y}_{opt}|)$ term is due to the deployment of the $|\mathcal{Y}_A|$ RNs. Note that $\Theta(|\mathcal{Y}_{opt}|)$ time is required for any algorithm to deploy the RNs. If one is only interested in the 2-connected spanning subgraph G_A, $O(n^4)$ time is sufficient. ■

C. Approximation Algorithm for 1tFTPB$(r, R, \mathcal{B}, \mathcal{X})$

We generalize the steinerized graph concept defined in Section 3-B to include BSs. Given a set of BSs $\mathcal{B}$, a set of SNs $\mathcal{X}$, and communication ranges $R \geq r > 0$, the 4-tuple $(r, R, \mathcal{B}, \mathcal{X})$ induces an edge weighted undirected complete graph $G^S(r, R, \mathcal{B}, \mathcal{X})$, called the *steinerized graph*, with vertex set $V = \mathcal{B} \cup \mathcal{X}$ and edge costs defined in the following way.

For two SNs x_i and x_j, the cost of edge (x_i, x_j) is defined by formula (3.1). For two BSs b_i and b_j, the cost of edge (b_i, b_j) is $c(b_i, b_j) = 0$. For a BS b_i and an SN x_j, the cost of edge (b_i, x_j) is defined as

$$c(b_i, x_j) = \begin{cases} 0, & \text{if } d(b_i, x_j) \in [0, r]; \\ \lceil \frac{d(b_i, x_j) - r}{R} \rceil, & \text{otherwise.} \end{cases} \quad (3.5)$$

We have assumed that the transmission range of BSs is big enough ($\gg R$) so that any two BSs are connected without the aid of RNs or SNs. Then, $c(z_i, z_j)$ is the minimum number of RNs needed to connect nodes $z_i, z_j \in \mathcal{B} \cup \mathcal{X}$ without the aid of any other nodes. The *steinerization* of SN-SN edges defined in Section 3-B can be generalized naturally to the case of BS-SN edges (based on formula (3.5)). The steinerization of a BS-BS edge does not deploy any RN. Our approximation algorithm for 1tFTPB is almost identical to that for 1tFTP except we also have to deal with BSs.

Algorithm 2 Approximation Algorithm for 1tFTPB

Input: $R \geq r > 0$, BSs $\mathcal{B}$ and SNs $\mathcal{X}$.
Output: RNs $\mathcal{Y}_A = \{y_1, y_2, \ldots, y_l\}$.

Construct the steinerized graph $G^S(r, R, \mathcal{B}, \mathcal{X})$.
Compute a 2-connected spanning subgraph G_A of $G^S(r, R, \mathcal{B}, \mathcal{X})$ using the algorithm $\mathcal{A}$ of [15].
$l := 0$;
for each edge $(z_i, z_j) \in G_A$ s.t. $c(z_i, z_j) \geq 1$ **do**
  Steinerize edge (z_i, z_j) with $c(z_i, z_j)$ relay nodes: $y_{l+1}, y_{l+2}, \ldots, y_{l+c(z_i, z_j)}$;
  $l := l + c(z_i, z_j)$.
end for

The concepts *layout*, *length of a layout*, and *shortest layout* defined in Definitions 3.4–3.5 generalize naturally to the case in the presence of BSs, and the *shortest layout of single tiered fault-tolerant relay node placement with basestations* for $(r, R, \mathcal{B}, \mathcal{X})$ is denoted by S1tFTLB. $\delta_b(y, \mathcal{L})$, the *basestation degree* of an RN y in a layout $\mathcal{L} = \mathcal{L}(\mathcal{Y})$ of an F1tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$, is the number of BSs adjacent with y in $\mathcal{L}$.

Lemma 3.1 and Lemma 3.2 can be generalized to the case with BSs, and proved verbatim. We list them (proof omitted) as Lemma 3.4 and Lemma 3.5 in the following.

Lemma 3.4: The RNs $\mathcal{Y}_A$ placed by Algorithm 2 is an F1tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$. Let $\mathcal{Y}_B$ be any F1tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$ that is obtained by steinerizing the edges of a 2-connected spanning subgraph of $G^S(r, R, \mathcal{B}, \mathcal{X})$. Then $|\mathcal{Y}_A| \leq 2 \cdot |\mathcal{Y}_B|$. ■

Lemma 3.5: Let $\mathcal{L} = \mathcal{L}(\mathcal{Y})$ be an S1tFTLB for $(r, R, \mathcal{B}, \mathcal{X})$. Then $\delta_s(y, \mathcal{L}) \leq 5$ for any RN $y \in \mathcal{Y}$. ■

We need the following lemma, which bounds the basestation degree of an RN in an S1tFTLB.

Lemma 3.6: Let $\mathcal{L}_S = \mathcal{L}(\mathcal{Y}_{min})$ be an S1tFTLB for $(r, R, \mathcal{B}, \mathcal{X})$, where $\mathcal{Y}_{min}$ is an M1tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$. Then $\delta_b(y, \mathcal{L}) \leq 1$ for any RN $y \in \mathcal{Y}_{min}$. □

PROOF. If there is at most one BS ($|\mathcal{B}| \leq 1$), the lemma is trivially true. Therefore we assume $|\mathcal{B}| \geq 2$ in the rest of this proof. Also, we assume that $\mathcal{L}_S$ contains all BS–BS edges, as addition of any such edge does not increase the length of the layout.

Let G' be a spanning subgraph of $HCG(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{min})$ such that G' *contains all of the* BS–BS *edges*. We claim that the following three statements are equivalent.

(1) G' is 2-connected.
(2) For any SN $x \in \mathcal{X}$ and any two BSs $b_i, b_j \in \mathcal{B}$, G' contains a cycle going through edge (b_i, b_j) and x.
(3) For any SN $x \in \mathcal{X}$, there exist two BSs $b_i, b_j \in \mathcal{B}$ such that G' contains a cycle going through (b_i, b_j) and x.

(1)⇒(2) : By our assumption, G' contains all the BS–BS edges. Therefore (b_i, b_j) is an edge in G' and x is a node in G'. Since G' is 2-connected, it contains a cycle going through edge (b_i, b_j) and node x.
(2)⇒(3) : This is trivially true.
(3)⇒(1) : By assumption, $\mathcal{X}$ contains at least one SN x_1. It follows from (3) that there exist two BSs b_i, b_j such that G' contains a cycle going through edge (b_i, b_j) and node x_1. Therefore b_i and b_j are on the same biconnected component of G'.

By our assumption, G' contains all BS–BS edges. Therefore all BSs are on the same biconnected component of G' (see Expansion Lemma in [25], p. 162).

For each SN $x \in \mathcal{X}$, (3) implies that G' contains a cycle going through two BSs in $\mathcal{B}$ and SN x. This implies that x is in the same biconnected component of G' with all the BSs. Therefore all BSs and SNs are in the same biconnected component of G'.

Note that $\mathcal{Y}_{min}$ is an M1tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$ and G' is a spanning subgraph of $HCG(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}_{min})$. Using the coloring scheme that we used near the end of the proof of Lemma 3.2, we conclude that G' is 2-connected (otherwise we would contradict the *minimum size* property of $\mathcal{Y}_{min}$). This proves the equivalence of the three statements (1)-(3).

Following the *minimum size* property of $\mathcal{Y}_{min}$ and the 2-connectivity of $\mathcal{L}_S$, we can prove the following claim.

(a): *For each* RN $y \in \mathcal{Y}_{min}$, *there exists an* SN $x \in \mathcal{X}$ *such that for any two* BS*s* $b_i, b_j \in \mathcal{B}$ *and any cycle* $\mathcal{C}$ *in* $\mathcal{L}_S$ *that uses edge* (b_i, b_j) *and node* x, $\mathcal{C}$ *must also contain node* y.

Assume that claim (a) is false. Then there exists an RN $y' \in \mathcal{Y}_{min}$ such that *for any* SN $x \in \mathcal{X}$, G' *contains a cycle* $\mathcal{C}$ *that goes through* SN x *and a* BS–BS *edge* (b', b''), *but not* RN y'. Let G'' be obtained from G' by deleting node y' and all edges adjacent with y'. Then G'' is a spanning subgraph of $HCG(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}'_{min})$, where $\mathcal{Y}'_{min} = \mathcal{Y}_{min} \setminus \{y'\}$. Since we have proved that (3)⇒(1), G'' is a 2-connected spanning subgraph of $HCG(r, R, \mathcal{B}, \mathcal{X}, \mathcal{Y}'_{min})$, contradicting the minimum size property of $\mathcal{Y}_{min}$. Therefore (a) is true.

Assume to the contrary that there exists an RN y such that $\delta_b(y, \mathcal{L}_S) \geq 2$. WLOG, assume that BSs b_i and b_j are adjacent with y in $\mathcal{L}_S$. It follows from claim (a) in the above that there exists an SN $x \in \mathcal{X}$ such that any cycle $\mathcal{C}$ of $\mathcal{L}_S$ that contains edge (b_i, b_j) and node x must also contain node y. Note that the existence of a cycle $\mathcal{C}$ of $\mathcal{L}_S$ which uses edge (b_i, b_j) and node x follows from the fact that $\mathcal{L}_S$ is 2-connected and contains all the BS–BS edges.

The cycle $\mathcal{C}$ can only take one of the following two forms: $x \to \cdots \to y \to b_i \to b_j \to \cdots \to x$, or $x \to \cdots \to b_i \to b_j \to y \to \cdots \to x$. If $\mathcal{C}$ has the first form, $\mathcal{L}_S$ contains the cycle $\mathcal{C}$ and its chord (y, b_j), contradicting the shortest length property of $\mathcal{L}_S$. If $\mathcal{C}$ has the second form, $\mathcal{L}_S$ contains the cycle $\mathcal{C}$ and its chord (y, b_i), again contradicting the shortest length property of $\mathcal{L}_S$. Therefore we have $\delta_b(y, \mathcal{L}_S) \leq 1$ for every RN y. ■

The concept of *Steiner components* defined in Definition 3.7 generalizes naturally to the case with BSs, with BS nodes treated similarly as sensor nodes. Similar to Lemma 3.3 and Theorem 3.1, we can prove the following (noting that $\delta_b(y, \mathcal{L}) + \delta_s(y, \mathcal{L}) \leq 6$ for each RN y involved).

Lemma 3.7: There exists a 2-connected spanning subgraph G_{appx} of the steinerized graph $G^S(r, R, \mathcal{B}, \mathcal{X})$ such that $c(G_{appx})$ is at most 8 times the size of an M1tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$. ■

Theorem 3.2: Algorithm 2 is a 16-approximation algorithm for 1tFTPB. It can compute a 2-connected spanning subgraph G_A of $G^S(r, R, \mathcal{B}, \mathcal{X})$ such that $c(G_A) \leq 16 \cdot |\mathcal{Y}_{opt}|$ in $O((m+n)^4)$ time, and requires additional $O((m+n)^2 + |\mathcal{Y}_{opt}|)$ time to deploy the RNs, where $\mathcal{Y}_{opt}$ is an M1tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$.■

We wish to point out that the algorithm of Lin and Xue [17], the algorithm of Lloyd and Xue [19], the algorithm of Kashyap *et al.* [14], and Algorithms 1 and 2 of this paper all follow the same three-stage design principle: STAGE 1: Construct the *steinerized graph* $G^S(r, R, \mathcal{B}, \mathcal{X})$ ($\mathcal{B}$ could be $\emptyset$); STAGE 2: Compute either an optimal solution (for k=1) or an approximation (for k=2) of the minimum cost k-connected spanning subgraph G_A of $G^S(r, R, \mathcal{B}, \mathcal{X})$ (k=1 for [17, 19] and k=2 for [14] and this paper); STAGE 3: Steinerize the edges of G_A. The difference lies in the analysis of the algorithms and in the computation of G_A.

For $k = 1$, the minimum cost 1-connected spanning subgraph of $G^S(r, R, \mathcal{B}, \mathcal{X})$ is the minimum spanning tree, which can be computed efficiently. For $k = 2$, computing the minimum cost 2-connected spanning subgraph of $G^S(r, R, \mathcal{B}, \mathcal{X})$ is NP-hard [15]. Therefore we compute a 2-approximation G_A using the algorithm of [15]. Although we don't have a theoretical proof, it has been observed in [14] that the Traveling Salesman (TSP) tour of G^S provides a 2-connected spanning subgraph of G^S whose cost is often very close to that of the minimum cost 2-connected spanning subgraph of G^S. Therefore a good heuristic algorithm is to use a TSP tour as a candidate for G_A, instead of using a 2-approximation to the minimum cost 2-connected spanning subgraph.

4. TWO-TIERED FAULT-TOLERANT RELAY PLACEMENT

A. Problem Definitions and Summary of Results

Two-Tiered Relay Node Placement with Basestations:

Definition 4.1: Let $\mathcal{B}$ be a set of BSs, $\mathcal{Y}$ be a set of RNs, and $R > 0$ be the communication range of the RNs. The *relay communication graph* $RCG(R, \mathcal{B}, \mathcal{Y})$ induced by the 3-tuple $(R, \mathcal{B}, \mathcal{Y})$ is an edge-weighted undirected graph with vertex set $V = \mathcal{B} \cup \mathcal{Y}$ and edge set E defined as follows:

- For any two BSs $b_i, b_j \in \mathcal{B}$, E contains the undirected edge $(b_i, b_j) = (b_j, b_i)$, with *length* $l(b_i, b_j) = 0$.
- For an RN $y_i \in \mathcal{Y}$ and a node $z_j \in \mathcal{Y} \cup \mathcal{B}$ which could be either an RN or a BS, E contains the undirected edge

$(y_i, z_j) = (z_j, y_i)$ if and only if $d(y_i, z_j) \le R$. The length of edge $(y_i, z_j) \in E$ is $l(y_i, z_j) = d(y_i, z_j)$.

The edge length function generalizes naturally to the *length of a subgraph* of RCG by summation. □

Definition 4.2: Let $r > 0$ and $R \ge r$ be the respective communication ranges for SNs and RNs. Let $\mathcal{B}$ be a set of BSs, and $\mathcal{X}$ be a set of SNs. A set of RNs $\mathcal{Y}$ is said to be a *feasible two-tiered fault-tolerant relay node placement with basestations* (denoted by F2tFTPB) for $(r, R, \mathcal{B}, \mathcal{X})$ if:

- For each SN $x \in \mathcal{X}$, there exist two RNs in $\mathcal{Y}$ that are within distance r of x.
- The graph $RCG(R, \mathcal{B}, \mathcal{Y})$ is 2-connected.

The *size* of the corresponding F2tFTPB is $|\mathcal{Y}|$. An F2tFTPB is a *minimum two-tiered fault-tolerant relay node placement with basestations* for $(r, R, \mathcal{B}, \mathcal{X})$ (denoted by M2tFTPB) if it has the minimum size among all F2tFTPBs for $(r, R, \mathcal{B}, \mathcal{X})$. The *two-tiered fault-tolerant relay node placement with basestations problem* for $(r, R, \mathcal{B}, \mathcal{X})$, denoted by 2tFTPB$(r, R, \mathcal{B}, \mathcal{X})$, seeks an M2tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$. □

Two-Tiered Relay Node Placement without Basestations: We also study a special case of 2tFTPB where $\mathcal{B} = \emptyset$. In this case, the relay communication graph defined above becomes $RCG(R, \mathcal{Y})$. Similarly,

- the term *feasible two-tiered fault-tolerant relay node placement with basestations* (F2tFTPB) for $(r, R, \mathcal{B}, \mathcal{X})$ becomes *feasible two-tiered fault-tolerant relay node placement* (F2tFTP) for $(r, R, \mathcal{X})$;
- the term *minimum two-tiered fault-tolerant relay node placement with basestations* (M2tFTPB) for $(r, R, \mathcal{B}, \mathcal{X})$ becomes *minimum two-tiered fault-tolerant relay node placement* (M2tFTP) for $(r, R, \mathcal{X})$;
- the term *two-tiered fault-tolerant relay node placement with basestations problem* (2tFTPB) for $(r, R, \mathcal{B}, \mathcal{X})$ becomes *two-tiered fault-tolerant relay node placement problem* (2tFTP) for $(r, R, \mathcal{X})$.

Discussions:
To our best knowledge, the 2tFTPB$(r, R, \mathcal{B}, \mathcal{X})$ problem has not been studied in the literature. Its special case, the 2tFTP$(r, R, \mathcal{X})$ problem, has been well studied.

Hao *et al.* in [9] first studied the 2tFTP problem under the name *two connected double cover*. The problem is conjectured to be NP-hard in both [9] and [23]. Approximation algorithms were presented in [9, 23, 18]. The approximation algorithm in [9] does not have a constant approximation ratio. The approximation algorithm in [23] has an approximation ratio of 4.5, provided that the SNs are uniformly distributed and that $R \ge 4r$. Without these conditions, the algorithm of [23] does not have any known approximation ratio. The approximation algorithm of [18] assumes $R = r$, but no assumption on the distribution of the SNs, and has an approximation ratio of $(24 + \epsilon)$, where $\epsilon > 0$ is any given constant.

The 2tFTPB problem is a more realistic model than the well-studied 2tFTP problem because a WSN is usually connected to one or more BSs and that BSs are more powerful than the SNs and RNs. To our knowledge, this is the first paper which studies the 2tFTPB problem.

Results:
We present a polynomial time $(10 + \epsilon)$-approximation algorithm for 2tFTP, improving the previous-best $(24 + \epsilon)$-approximation algorithm of [18] which was designed for the special case where $R = r$. We also present a polynomial time $(20 + \epsilon)$-approximation algorithm for 2tFTPB.

B. Approximation Algorithm for 2tFTP

In a recent paper [19], Lloyd and Xue presented a polynomial time $(5 + \epsilon)$-approximation algorithm for the following two-tiered relay node placement problem.

Definition 4.3: Let $R \ge r > 0$ be the respective communication ranges for RNs and SNs. Let $\mathcal{X}$ be a set of SNs and $\mathcal{Y}$ be a set of RNs. $\mathcal{Y}$ is said to be a *feasible two-tiered relay node placement* (denoted by F2tRNP) for $(r, R, \mathcal{X})$ if:

- For each SN $x \in \mathcal{X}$, there exists an RN $y \in \mathcal{Y}$ that is within distance r of x.
- The graph $RCG(R, \mathcal{Y})$ is connected.

The *size* of the corresponding F2tRNP is $|\mathcal{Y}|$. An F2tRNP is said to be a *minimum two-tiered relay node placement* for $(r, R, \mathcal{X})$ (denoted by M2tRNP) if it has the minimum size among all F2tRNPs for $(r, R, \mathcal{X})$. The *two-tiered relay node placement problem* for $(r, R, \mathcal{X})$, denoted by 2tRNP$(r, R, \mathcal{X})$, seeks an M2tRNP for $(r, R, \mathcal{X})$.

We use the result of Lloyd and Xue [19] in the design of the following $(10 + \epsilon)$-approximation algorithm for 2tFTP.

Algorithm 3 $(10 + \epsilon)$-Approximation Algorithm for 2tFTP

Input: $R \ge r > 0$, $\epsilon > 0$, sensor nodes $\mathcal{X} = \{x_1, \ldots, x_n\}$.
Output: A set of relay nodes $\mathcal{Y}_A = \{y_1, \ldots, y_l\}$.
1: Apply algorithm $\mathcal{A}$ of [19] to compute a $(5 + \epsilon/2)$-approximation to 2tRNP$(r, R, \mathcal{X})$, given by the set of relay nodes $\mathcal{Z} = \{z_1, \ldots, z_k\}$.
2: Duplicate each of the relay nodes in $\mathcal{Z}$ to obtain $\mathcal{Y}_A$.

Theorem 4.1: For any given constant $\epsilon > 0$, Algorithm 3 computes a $(10 + \epsilon)$-approximation to 2tFTP$(r, R, \mathcal{X})$ in polynomial time.

PROOF. The polynomial running time of Algorithm 3 follows that of the $(5 + \epsilon)$-approximation algorithm for 2tRNP [19].

Since $\mathcal{Z}$ is an F2tRNP for $(r, R, \mathcal{X})$, we know that each SN $x \in \mathcal{X}$ is within distance r of an RN $z \in \mathcal{Z}$, and that the relay communication graph $RCG(R, \mathcal{Z})$ is connected. By the construction of $\mathcal{Y}_A$, we know that each SN $x \in \mathcal{X}$ is within distance r of two RNs $y, y' \in \mathcal{Y}_A$, and that the relay communication graph $RCG(R, \mathcal{Y}_A)$ is 2-connected.

Let $\mathcal{Z}_{opt}$ be an M2tRNP for $(r, R, \mathcal{X})$ and $\mathcal{Y}_{opt}$ be an M2tFTP for $(r, R, \mathcal{X})$. It follows from the definitions of 2tRNP and 2tFTP, any F2tFTP for $(r, R, \mathcal{X})$ is guaranteed to be an F2tRNP for $(r, R, \mathcal{X})$. This implies $|\mathcal{Z}_{opt}| \le |\mathcal{Y}_{opt}|$. Since $\mathcal{A}$ is a $(5 + 0.5\epsilon)$-approximation algorithm for 2tRNP, we have $|\mathcal{Z}| \le (5 + 0.5\epsilon) \times |\mathcal{Z}_{opt}|$. It follows that

$$|\mathcal{Y}_A| = 2|\mathcal{Z}| \le (10 + \epsilon) \times |\mathcal{Z}_{opt}| \le (10 + \epsilon) \times |\mathcal{Y}_{opt}|. \quad (4.1)$$

Therefore Algorithm 3 is a $(10+\epsilon)$-approximation algorithm.■

C. Approximation Algorithm for 2tFTPB

Our approximation algorithm for 2tFTPB uses an PTAS for the NP-hard DCover problem defined in the following [11].

Definition 4.4: Let $\mathcal{X}$ be a set of points in the Euclidean plane, and let $r > 0$ be a positive constant. A set of points $\mathcal{D}$ is said to be a *geometric disk cover* of $\mathcal{X}$ if for each point $x_i \in \mathcal{X}$ there exists a point $d_j \in \mathcal{D}$ s.t. $d(x_i, d_j) \le r$. The *minimum geometric disk cover* problem for $(r, \mathcal{X})$ (denoted by DCover$(r, \mathcal{X})$) seeks a minimum cardinality cover of $\mathcal{X}$. □

Algorithm 4 $(20+\epsilon)$-Approximation Algorithm for 2tFTPB

Input: $R \ge r > 0$, $\epsilon > 0$, SNs $\mathcal{X}$, BSs $\mathcal{B}$.
Output: A set of RNs $\mathcal{Y}_A \cup \mathcal{D}_A \cup \mathcal{U}_A$.

1: Apply algorithm $\mathcal{A}$ of [11] to obtain a minimal set of RNs $\mathcal{D}$ which is a $(1+0.25\epsilon)$-approximation to DCover$(r, \mathcal{X})$.
2: Construct a set $\mathcal{U} \subseteq \mathcal{X}$ s.t $\forall\ d_i \in \mathcal{D}$, $\exists$ exactly one $u_j \in \mathcal{U}$ with $d(d_i, u_j) \le r$ and that $\forall\ u_i \in \mathcal{U}$, $\exists$ exactly one $d_j \in \mathcal{D}$ with $d(u_i, d_j) \le r$. {Note that $|\mathcal{U}| = |\mathcal{D}|$.}
3: Apply Algorithm 2 to obtain a set of RNs $\mathcal{Y}_A$ that is a 16-approximation to 1tFTPB$(R, R, \mathcal{B}, \mathcal{U})$.
4: Duplicate each RN in $\mathcal{D}$ to obtain $\mathcal{D}_A$. Duplicate each RN in $\mathcal{U}$ to obtain $\mathcal{U}_A$.

Theorem 4.2: For any given constant $\epsilon > 0$, Algorithm 4 computes a $(20+\epsilon)$-approximation to 2tFTPB$(r, R, \mathcal{X})$ in polynomial time. □

PROOF. Lines 1 and 3 each invokes a polynomial time algorithm. Lines 2 and 4 each takes polynomial time (in both the input size and the output size). Therefore Algorithm 4 is a polynomial time algorithm.

Since $\mathcal{D}$ is a geometric disk cover of $\mathcal{X}$, and $\mathcal{D}_A$ is obtained by duplicating each RN in $\mathcal{D}$, each SN in $\mathcal{X}$ is within distance r of at least two RNs in $\mathcal{D}_A$. Since $\mathcal{Y}_A$ is an F1tFTPB for $(R, R, \mathcal{B}, \mathcal{U})$, all RNs in $\mathcal{Y}_A$ and $\mathcal{U}$ and the BSs in $\mathcal{B}$ are in a common biconnected component of $RCG(R, \mathcal{B}, \mathcal{Y}_A \cup \mathcal{U})$. Following the construction of $\mathcal{U}$, each RN in $\mathcal{D}$ is within distance $r \le R$ of an RN in $\mathcal{U}$. Since $\mathcal{D}_A$ has two copies of each RN in $\mathcal{D}$ and $\mathcal{U}_A$ has two copies of each RN in $\mathcal{U}$, $RCG(R, \mathcal{B}, \mathcal{Y}_A \cup \mathcal{U}_A \cup \mathcal{D}_A)$ is 2-connected. This proves that $\mathcal{Y}_A \cup \mathcal{U}_A \cup \mathcal{D}_A$ is an F2tFTPB for $(r, R, \mathcal{B}, \mathcal{X})$.

Let $\mathcal{Y}_{opt}$ be an optimal solution for 2tFTPB$(r, R, \mathcal{B}, \mathcal{X})$, $\mathcal{D}_{min}$ be an optimal solution for DCover$(r, \mathcal{X})$, and $\mathcal{Y}_{sub}$ be an optimal solution for 1tFTPB$(R, R, \mathcal{B}, \mathcal{U})$. Since $\mathcal{Y}_{opt}$ is also a geometric disk cover of $\mathcal{X}$ (with radius r), we have $|\mathcal{D}_{min}| \le |\mathcal{Y}_{opt}|$. Since $\mathcal{Y}_{opt}$ is a feasible solution for 2tFTPB$(r, R, \mathcal{B}, \mathcal{X})$, $\mathcal{U}$ is placed on a subset of $\mathcal{X}$ and $R \ge r$, we conclude that $\mathcal{Y}_{opt}$ is a feasible solution for 2tFTPB$(R, R, \mathcal{B}, \mathcal{U})$, which in turn implies that $\mathcal{Y}_{opt}$ is a feasible solution for 1tFTPB$(R, R, \mathcal{B}, \mathcal{U})$. Since $\mathcal{Y}_{sub}$ is an optimal solution for 1tFTPB$(R, R, \mathcal{B}, \mathcal{U})$, we have $|\mathcal{Y}_{sub}| \le |\mathcal{Y}_{opt}|$. Following the algorithm, we have

$$|\mathcal{D}_A| = 2|\mathcal{D}| \le 2(1+0.25\epsilon)\mathcal{D}_{min} \le (2+0.5\epsilon)|\mathcal{Y}_{opt}|,$$
$$|\mathcal{U}_A| = 2|\mathcal{U}| = 2|\mathcal{D}| \le (2+0.5\epsilon)|\mathcal{Y}_{opt}|,$$
$$|\mathcal{Y}_A| \le 16|\mathcal{Y}_{opt}|.$$

Therefore, we have

$$|\mathcal{Y}_A \cup \mathcal{D}_A \cup \mathcal{U}_A| \le (20+\epsilon) \cdot |\mathcal{Y}_{opt}|. \tag{4.2}$$

This completes the proof of Theorem 4.2. ■

Line 4 of Algorithm 4 is very pessimistic, but leads to a shorter proof of Theorem 4.2. All claims in Theorem 4.2 are still true if Line 4 of Algorithm 4 is replaced by the following.

4A: Set $\mathcal{D}_A := \mathcal{D}$ and $\mathcal{U}_A := \mathcal{U}$.
4B: For each RN in $\mathcal{D}_A$ that is an *articulation point* of $RCG(R, \mathcal{B}, \mathcal{Y}_A \cup \mathcal{D}_A \cup \mathcal{U}_A)$, add a duplicate of it to $\mathcal{D}_A$.
4C: For each SN $x \in \mathcal{X}$ that is within distance r of exactly one RN in $\mathcal{Y}_A \cup \mathcal{D}_A \cup \mathcal{U}_A$, add a duplicate of that RN.

Due to space limitations, we omit the proof of this claim.

5. Numerical Results

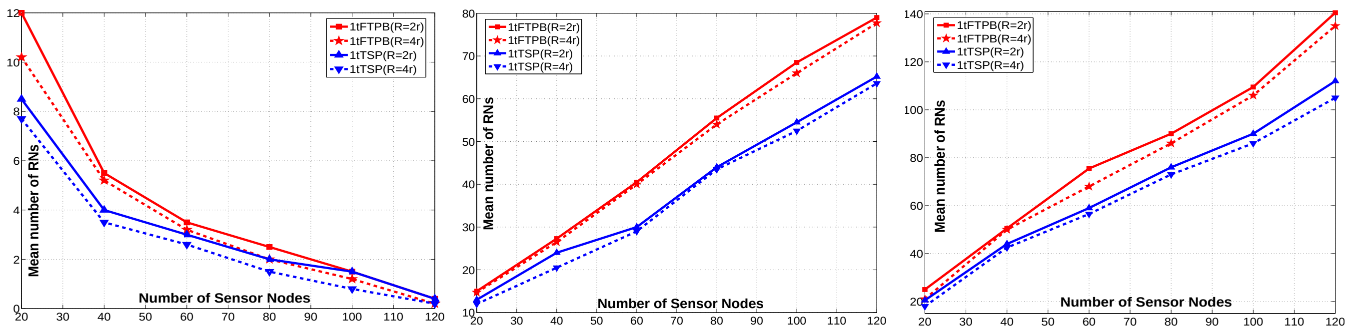
We have carried out computational studies of our algorithms on randomly generated test problems. We present results for 1tFTPB and 2tFTPB, as 1tFTP and 2tFTP can be viewed as special cases. In the figures and discussions, we will use 1tFTPB and 2tFTPB also to denote our corresponding algorithms for solving the problems. Our algorithm for 2tFTPB uses the PTAS of [11] for computing a $(1+0.25\epsilon)$-approximation for DCover. The PTAS uses an integer parameter $\ell > 0$ and *guarantees* a $(1+\frac{1}{\ell})^2$-approximation. We found that the results *produced* by the algorithm were almost the same with ℓ set to 3 and to 2. So we fixed $\ell = 2$ in our implementations, yielding a *guaranteed* 25-approximation.

Since there are no previous algorithms for solving these problems, and that optimal solutions are difficult to obtain, we also implemented two heuristics: 1tTSP and 2tTSP. 1tTSP computes a TSP tour of $G^S(r, R, \mathcal{B}, \mathcal{X})$ and steinerizes the edges of the tour to deploy RNs. As discussed near the end of Section 3, 1tTSP may produce close to optimal solutions. 2tTSP is similar to 1tTSP, but computes a TSP tour of $G^S_+(r, R, \mathcal{B}, \mathcal{X})$, which is the same as $G^S(r, R, \mathcal{B}, \mathcal{X})$ except that SN-SN edges that have length 0 in G^S now have length 1 in G^S_+ (due to the covering need). When the SNs are sparsely distributed, the length of the computed TSP tour is close to the number of RNs needed to steinerize the edges of an Euclidean TSP tour of the nodes. Therefore 2tTSP should produce close to optimal solutions in this case. **Note that neither 1tTSP nor 2tTSP is a polynomial time algorithm.** We have used the Concorde TSP Solver [28] to compute TSP tours.

As in [14] and [23], SNs $\mathcal{X}$ were uniformly distributed in a square playing field. Two basestations were randomly deployed in the square. Fig. 4 and 5 illustrate the average of 10 test runs for various scenarios. First, we study the scenario where the number of SNs increases but the playing field is fixed at 100×100 sq. units. As expected, the number of RNs needed for 1tFTPB decreases with n, and converges to 0, (see Fig. 4(a)); the number of RNs needed for 2tFTPB increases with n, and converges to a constant, (see Fig. 5(a)).

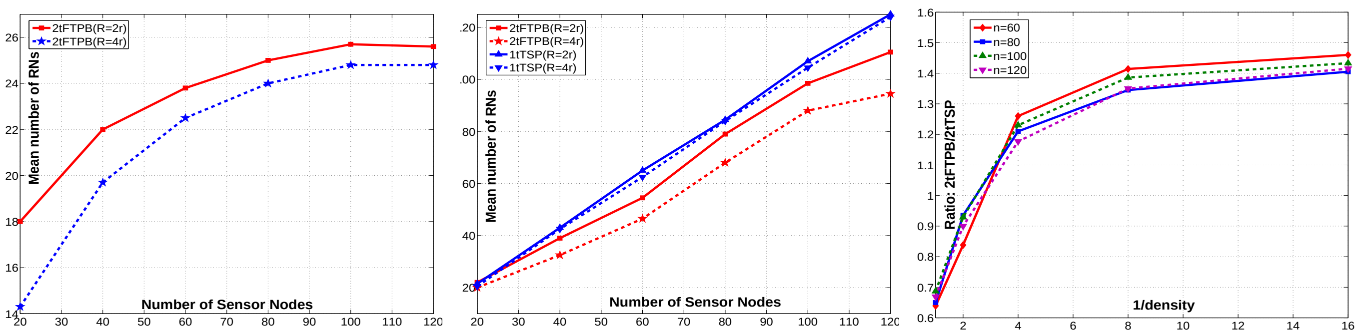
Fig. 4(b) and Fig. 4(c) illustrate the growth of the number of RNs needed for 1tFTPB as n increases while the playing field also grows to keep the sensor density constant, where *density* is the number of SNs in one square unit. Fig. 5(b) illustrates the case for 2tFTPB. As expected, the number of RNs needed grows almost linearly with n in both cases.

Fig. 5(c) shows the ratio of the number of RNs needed by Algorithm 4 over that needed by 2tTSP, as a function of the reciprocal of the sensor density. As discussed earlier, the 2tTSP heuristic produces close to optimal solutions when the sensor density becomes very small. Fig. 5(c) shows that the number of RNs required by Algorithm 4 is no more than 1.5 times the number of RNs required by 2tTSP. This suggests that Algorithm 4 has very good performance.



(a) 100×100 playing field, $r = 20$ (b) density = 0.50 (c) density = 0.25

Fig. 4. Numerical Results for 1tFTPB: 1tFTPB also denotes the algorithm, and is compared with 1tTSP.



(a) 100×100 playing field, $r = 20$ (b) density = 0.50 (c) 2tFTPB/2tTSP ratio

Fig. 5. Numerical Results for 2tFTPB: 2tFTPB also denotes the algorithm, and is compared with 2tTSP.

6. Conclusions

We have studied four fault-tolerant relay node placement problems in WSNs and presented an $O(1)$-approximation algorithm for each of them. These problems have been previously studied only in restricted forms. Besides improving/generalizing previously best known results, our paper has the following features. We provide fault-tolerance (compared with 1-connectivity) and maintain simplicity (compared with higher order connectivity). We distinguish the communication powers of the basestations, the relay nodes, and the sensor nodes. We also explore the two-tiered network architecture.

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; *IEEE Globecom'2004*; pp. 3210–3214.

[5] X. Cheng, D.Z. Du, L. Wang and B. Xu; Relay sensor placement in wireless sensor networks; *ACM/Springer WINET*; accepted. Available at http://www.seas.gwu.edu/˜cheng/Publication/relay.pdf.

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

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

[8] G. Gupta and M. Younis; Fault tolerant clustering of wireless sensor networks; *WCNC'03*, pp. 1579–1584.

[9] B. Hao, J. Tang and G. Xue; Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation; *IEEE HPSR'04: Workshop on High Performance Switching and Routing*; pp. 246-250.

[10] W.B. Heinzelman, A.P. Chandrakasan and H. Balakrishnan; An application-specific protocol architecture for wireless microsensor networks; *IEEE Trans. Wireless Comm.*; Vol. 1(2002), pp. 660–670.

[11] D.S. Hochbaum and W. Maass; Approximation schemes for covering and packing problems in image processing and VLSI; *Journal of the ACM*; Vol. 32(1985), pp. 130–136.

[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] B. Karp and H. Kung; GPSR: greedy perimeter stateless routing for wireless networks; *MobiCom'2000*, pp. 243–254.

[14] A. Kashyap, S. Khuller and M. Shayman; Relay placement for higher order connectivity in wireless sensor networks; *Infocom'2006*.

[15] S. Khuller and B. Raghavachari; Improved approximation algorithms for uniform connectivity problems; *J. of Alg.*; Vol. 21(1996), pp. 214–235.

[16] Q. Li, J. Aslam, D. Rus; Online power-aware routing in wireless ad-hoc networks; *MobiCom'2001*, pp. 97–107.

[17] 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.

[18] 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.

[19] E. Lloyd and G. Xue; Relay node placement in wireless sensor networks; *IEEE Transactions on Computers*; accepted for publication, available at http://www.fulton.asu.edu/˜xue/papers/TC2006-RelayNode.pdf.

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

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

[22] A. Srinivas, E. Modiano; Minimum energy disjoint path routing in wireless ad-hoc network; *MobiCom'2003*, pp. 122–133.

[23] J. Tang, B. Hao and A. Sen; Relay node placement in large scale wireless sensor networks; *Comput. Communications*; Vol. 29(2006), pp. 490–501.

[24] W. Wang, V. Srinivansan and K.C. Chua; Using mobile relays to prolong the lifetime of wireless sensor networks; *MobiCom'05*; pp. 270–283.

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

[26] 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.

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

[28] *The Concorde TSP Solver*, http://www.tsp.gatech.edu/concorde.html.