# Traffic-Aware Relay Node Deployment for Data Collection in Wireless Sensor Networks

Feng Wang*
School of Computing Science
Simon Fraser University
British Columbia, Canada
Email: fwa1@cs.sfu.ca

Dan Wang†
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Hong Kong
Email: csdwang@comp.polyu.edu.hk

Jiangchuan Liu*
School of Computing Science
Simon Fraser University
British Columbia, Canada
Email: jcliu@cs.sfu.ca

***Abstract*—Wireless sensor networks have been widely used for ambient data collection in diverse environments. While in many such networks the sensor nodes are randomly deployed in massive quantity, there is a broad range of applications advocating manual deployment. A typical example is structure health monitoring, where the sensors have to be placed at critical locations to fulfill civil engineering requirements. The raw data collected by the sensors can then be forwarded to a remote base station (the *sink*) through a series of *relay* nodes.**

**In the wireless communication context, the operation time of a battery-limited relay node depends on its traffic volume and communication range. Hence, although not bounded by the civil-engineering-like requirements, the locations of the relay nodes have to be carefully planned to achieve the maximum network lifetime. The deployment has to not only ensure connectivity between the data sources and the sink, but also accommodate the heterogeneous traffic flows from different sources and the dominating many-to-one traffic pattern.**

**Inspired by the uniqueness of such application scenarios, in this paper, we present an in-depth study on the traffic-aware relay node deployment problem. We develop optimal solutions for the simple case of one source node, both with single and multiple traffic flows. We show however that the general form of the deployment problem is difficult, and the existing connectivity-guaranteed solutions cannot be directly applied here. We then transform our problem into a generalized version of the Euclidean Steiner Minimum Tree problem (ESMT). Nevertheless, we face further challenges as its solution is in continuous space and may yield fractional numbers of relay nodes, where simple rounding of the solution can lead to poor performance. We thus develop algorithms for discrete relay node assignment, together with local adjustments that yield high-quality practical solutions. Our solution has been evaluated through both numerical analysis and $ns$-2 simulations and compared with state-of-the-art approaches. The results show that it achieves up to $6$ to $14$ times improvement on the network lifetime over the existing traffic-oblivious strategies.**

## I. Introduction

Wireless sensor networks have been widely used for ambient data collection in diverse environments. Examples include target-tracking [5] on battlefield and forest fire detection [15] in a wild environment, to name but a few. In many such networks, the sensor nodes are randomly deployed in massive quantities, and each node may act both as a data collector and a traffic relay. This is also a common assumption made in many existing works on modeling and protocol optimization, and the focus thus has been put on optimizing topology control [18][22][11][12] and routing design [20][4][13][3] with the given network topologies.

In contrast to this, we notice that there is another broad range of application scenarios that require manual node deployment. One example is the TsingMa Bridge [10] in Hong Kong, which is equipped with a large number of accelerometers, thermometers and strain sensors to monitor its working conditions. Another recent project, in which we are participating, is the Guangzhou New TV Tower [1] in Guangzhou, China, which is to be attached with similar sensors for real-time monitoring and analyzing. In these systems, the sensors are deployed at critical locations to fulfill civil engineering requirements. Raw data are needed and the traffic volume or data rate from each sensor is in general predetermined, e.g., the typical sampling rate of an accelerometer is $100Hz$. Given the extensive dimensions of the structures, relay nodes have to be placed to bridge the sensors and the data collection sink.

In the wireless communication scenario, the lifetime of a relay node is severely limited by its battery power, and the power consumption in turn closely depends on the communication distance and traffic volume. As such, although not bounded by the civil-engineering-like requirements, the locations of the relay nodes have to be carefully planned to achieve the best network performance.

Inspired by the uniqueness of these applications, in this paper, we present an in-depth study on the traffic-aware relay node deployment problem. There have been previous studies on relay node deployment for wireless networks [27][16], most of which however focused on maintaining network connectivity. Given the heterogenous traffic flows and the many-to-one traffic pattern, directly applying these algorithms will only give suboptimal results. For an illustration, consider a set of sensor nodes and a data sink with given locations and traffic volumes, as shown in Fig. 1. If only connectivity is considered, the deployment scheme in Fig. 1(a) maximizes the network lifetime, i.e., each of the $\frac{1}{3}$ of the relay nodes are deployed on the sections of $(s_1, v)$, $(s_2, v)$ and $(v, s_0)$. We can see, however, given the traffic pattern, moving some relay nodes

*Feng Wang and Jiangchuan Liu's work is supported by a Canada NSERC Discovery Grant and an NSERC Strategic Project Grant.

†Dan Wang's work is supported by HKPU/ICRG under Grant G-YG78, A-PB0R, RGC/GRF under Grant 5305/08E.

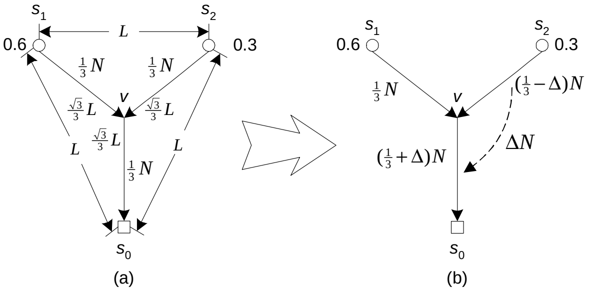


Fig. 1: An example of relay node deployment: (a) connectivity-only deployment; (b) traffic-aware deployment. $s_1$, $s_2$ are sources with data rate of $0.6$ and $0.3$. $s_0$ is the sink.

from less traffic intensive section $(s_2, v)$ to $(v, s_0)$ will achieve better performance, as shown in Fig. 1(b).

Facing the distinct traffic-aware demand, we first develop optimal solutions for the simple case of one source node, both with single and multiple traffic flows. We show that the general form of the deployment problem however is quite difficult. Indeed, even without traffic considerations, the relay node deployment problem is already NP-hard with heuristics being developed [27]. Unfortunately, our analysis shows that their approaches are far from optimized in our scenario. To this end, we show that the general problem can be transformed into a generalized Euclidian Steiner Minimum Tree problem (ESMT) and develop a hybrid algorithm that successfully returns optimal results with all test cases that can be verified within acceptable timeframes. Nevertheless, we face further challenges as the solution of ESMT is in the continuous domain and may yield fractional numbers of relay nodes. We show that a simple rounding of the solution may result in significant degradation of the performance. We then develop algorithms for discrete relay node assignment, together with local adjustments that yield high-quality practical solutions.

Our solutions have been evaluated using both numerical analysis and $ns$-2 simulations. We show that the performance of our scheme is $14$ times better than a straightforward relay node deployment that places the relay nodes in straight line to connect each source and the sink separately. Our scheme also outperforms by $6$ times than the state-of-the-art algorithm considering connectivity only.

The remaining part of the paper proceeds as follows. Section II presents the related work. We outline the system models and the problem description in Section III. Section IV proposes solutions to several case studies which can be used as building blocks for the general problem. In Section V, we study the general problem in-depth by first developing solutions in continuous space and then focusing on discrete deployment. We evaluate our solution by both numerical results and $ns$-2 simulations in Section VI. Finally Section VII concludes our paper and gives directions of future work.

## II. Background and Related Work

It is known that the energy of a sensor node is mainly consumed by the wireless communication, which is proportional to the data rate and the communication distance [6]. Since the latter is adjustable, many studies have explored this property to achieve topology control with given node deployment. In [18], an optimization problem is formulated to minimize the maximum power used for each individual node while maintaining the network connectivity. There are many follow-up efforts in this direction [22][11][12]. Another common goal is power-aware routing [20]. Given the traffic load, an integer programming can be formulated to minimize the maximum node energy consumption [4], where the data routes and the corresponding power levels are identified. Followup studies with different objectives or constraints can be found in [13][3]. They generally have assumed that the deployment of the network nodes is given, which often follows a random distribution.

Relay node deployment for WSNs has been studied in various contexts [27][16][24][7]. The connectivity problem for relay node deployment was first formulated in [14], and shown to be NP-hard. An approximation algorithm was then proposed based on *steinerization*, which assigns all relay nodes with roughly the same distance on each edge. This problem was generalized to $k$-connectivity in [2], which is also named as the survivability problem for $k \geq 2$. Later [16] further extended the problem by considering the constraint that relay nodes can only be placed at some given locations. On the other hand, there are several works [24][7] explicitly considering relay node placement to prolong network lifetime, e.g., [24] focuses on massive random relay node deployment and [7] stresses on using energy provisioning and giving each relay node different energy budget to achieve better performance. Our work, different from aforementioned, explicitly considers the unique traffic pattern in WSNs for data collection.

## III. System Model and Problem Description

We consider a wireless sensor network that consists of *source nodes* (or *S-nodes* in short) and *relay nodes* (or *R-nodes* in short). S-nodes sense the ambient environment and forward the data, through R-nodes, to a remote base station for further processing. The locations of S-nodes and the base station are given according to application requirements. The data rates of S-nodes are also known, but may be different for different S-node depending on the specific type of data sensed.

Given these application-specific conditions, the network lifetime thus closely depends on the geographical deployment of the R-nodes, as illustrated in Fig. 1. Let $S = \{s_1, s_2, ..., s_M\}$ denote the set of locations of $M$ S-nodes and $s_0$ be the location of the base station. Let the data rate from $s_i$ be $\gamma_i$. Define traffic path $p_i = x_0 x_1 \dots x_{l_i}$ as a sequence of R-nodes which participate in relaying the traffic flow from $s_i$. Similar to [24][7], we consider the problem how to deploy a given number of R-nodes so as to maximize the network lifetime, which can be formulated as follows:

**Traffic-Aware Deployment Problem**: Given $N$, the total number of R-nodes to be deployed, where $N \geq M$, find the geographical locations for R-nodes $F =$

$\{f_1, f_2, \ldots, f_N\}$, together with their respective communication ranges $R = \{r_1, r_2, \ldots, r_N\}$ and traffic paths for S-nodes $P = \{p_1, p_2, \ldots, p_M\}$, so as to minimize the energy consumption of the R-nodes. Specifically, since the network lifetime is critically bounded by the nodes with the highest energy costs, we are interested in minimizing the maximum energy consumption among the R-nodes, i.e.,

$$\min \max_{1 \le i \le N} \sum_{i \in p_j, j=1..M} \gamma_j [\mathbb{E}_{recv} + \mathbb{E}_{send}(r_i)] \ .$$

Notice that the summation here indicates that an R-node can undertake combined traffic flows of multiple sources if it is chosen in these paths. Denote $R_{max}$ is the maximum communication range of an R-node. The deployment should satisfy the following constraints:

(1) Communication range,

$$\forall r \in R, r \le R_{max};$$

(2) Forwarding path connectivity,

$$\forall p = x_0 x_1 \ldots x_l \in P, \overline{f_{x_{i-1}} f_{x_i}} \le r_{x_{i-1}}, i = 1 \ldots l;$$

(3) S-nodes and sink connectivity,

$$\forall s \in S, \exists p = x_0 x_1 \ldots x_l \in P, f_{x_0} = s, \overline{f_{x_l} s_0} \le r_{x_l}.$$

To simplify exposition, we associate each S-node with an R-node at the same location (as shown in Constraint 3), which guarantees S-nodes are only involved in local communications and the network lifetime thus depends on R-nodes.

Our formulation is not restricted by specific energy models for wireless communications. For illustration purpose, the following popular energy consumption model for packet transmission [17] will be used in this paper:

$$\mathbb{E}_{send}(r) = ar^{\alpha} + b \ ,$$

which can also be normalized as

$$\mathbb{E}_{send}(r) = r^{\alpha} + c \ ,$$

where $c$ is a small constant comparing with $r^{\alpha}$. The energy consumption for packet receiving is given by

$$\mathbb{E}_{recv} = c \ .$$

Finally, it is worth noting that our network model can be easily extended to a hierarchial structure where each S-node represents a cluster of geographically-close sources [7][19][23]. Our analysis and optimization below will still apply as long as the many-to-one pattern holds and the inter-cluster communications dominate the energy consumption, which is the case for most applications.

## IV. Optimal Traffic-Aware Relay Node Deployment: The Single Source Case

In this section, we study the relay deployment problem of two basic cases with single source, and derive optimal solutions. These results will serve as building blocks for solving the general problem in the next section.

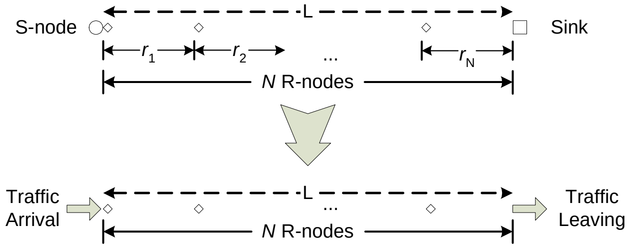
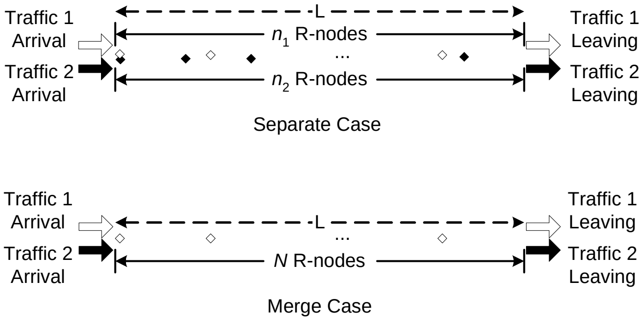


Fig. 2: An illustration of deployment for single source single flow and its generalization.



Separate Case

Merge Case

Fig. 3: An illustration of two deployment schemes for single source two traffic flows.

### A. The Single Source Single Traffic Flow Case

We begin with the basic case of single source single traffic flow. An illustration is shown in Fig. 2, where $L$ is the distance between the S-node $s_1$ (with traffic rate $\gamma_1$) and the sink $s_0$. We need to deploy $N$ R-nodes between them. Obviously, $N$ should satisfy $\frac{L}{N} \le R_{max}$ for a feasible solution. Let the distance between the $i$-th R-node and its next R-node/sink be $r_i$, $i = 1, \ldots, N$, the energy cost for the $i$-th R-node is

$$\gamma_1 [\mathbb{E}_{recv} + \mathbb{E}_{send}(r_i)] = 2\gamma_1 c + \gamma_1 r_i^{\alpha}.$$

Since $\sum_{i=1}^{N} r_i = L$, it is easy to see that the solution to $\min \max_{1 \le i \le N} (2\gamma_1 c + \gamma_1 r_i^{\alpha})$ is $r_i = \frac{L}{N}$, for $i = 1, 2, \ldots, N$, and the minimum of the maximum energy consumption among the R-nodes is $2\gamma_1 c + \gamma_1 (\frac{L}{N})^{\alpha}$. This result can be generalized as follows, which can be easily proved by contradiction.

*Theorem 1:* The optimal solution for single source single traffic flow is to start from the source and evenly deploy the R-nodes with an in-between distance of $\frac{L}{N}$. The energy consumption for each R-node is $\mathbb{E}_{single}(L, N, \gamma) = \gamma[2c + (\frac{L}{N})^{\alpha}]$.

### B. The Single Source Multi Traffic Flow Case

Next, we consider the case where multiple traffic flows arrive at one location and need to be relayed to another. Given $N$ R-nodes and $K$ traffic flows, we need to decide whether to merge these flows or to relay them separately by assigning $n_i$ R-nodes to the $i$-th flow, as long as $\sum_{i=1}^{K} n_i = N$. We first consider the case of two flows, which is illustrated in Fig. 3.

If the traffic flows are relayed separately, according to Theorem 1, the energy consumption of one R-node for the $i$-th traffic flow is $\mathbb{E}_{single}(L, n_i, \gamma_i)$, for $i = 1, 2$. Similar to the idea used in the previous subsection, it is easy to see that the R-nodes should be assigned such that

$$\mathbb{E}_{single}(L, n_1, \gamma_1) = \mathbb{E}_{single}(L, n_2, \gamma_2)\ .$$

Consequently,

$$\gamma_1[2c + (\frac{L}{n_1})^\alpha] = \gamma_2[2c + (\frac{L}{n_2})^\alpha]\ .$$

Typically, we have $(\frac{L}{n_i})^\alpha \gg c$ [8] for $i = 1, 2$ and thus

$$\gamma_1(\frac{L}{n_1})^\alpha \approx \gamma_2(\frac{L}{n_2})^\alpha\ ,$$

which follows

$$\frac{\sqrt[\alpha]{\gamma_1}L}{n_1} = \frac{\sqrt[\alpha]{\gamma_2}L}{n_2} = \frac{L(\sqrt[\alpha]{\gamma_1} + \sqrt[\alpha]{\gamma_2})}{n_1 + n_2} = \frac{L}{N}(\sqrt[\alpha]{\gamma_1} + \sqrt[\alpha]{\gamma_2})\ .$$

We then have the energy consumption of an R-node as

$$\mathbb{E}_{separarte} \approx (\frac{L}{N})^\alpha(\sqrt[\alpha]{\gamma_1} + \sqrt[\alpha]{\gamma_2})^\alpha\ .$$

On the other hand, if the traffic flows are merged, the energy consumption of one R-node becomes

$$\mathbb{E}_{merge} = \mathbb{E}_{single}(L, N, (\gamma_1 + \gamma_2)) \approx (\gamma_1 + \gamma_2)(\frac{L}{N})^\alpha\ .$$

Clearly, we have

$$\begin{aligned} \mathbb{E}_{separate} &= (\frac{L}{N})^\alpha(\sqrt[\alpha]{\gamma_1} + \sqrt[\alpha]{\gamma_2})^\alpha \\ &\geq (\frac{L}{N})^\alpha[(\sqrt[\alpha]{\gamma_1})^\alpha + (\sqrt[\alpha]{\gamma_2})^\alpha] = \mathbb{E}_{merge}\ , \end{aligned}$$

which shows that merging these two flows leads to the minimum energy cost on an R-node. This result can be easily generalized to the case of $K$ traffic flows [21] as follows:

*Theorem 2:* The optimal solution to single source multi traffic flow is to merge all flows into one and apply the optimal scheme of single source with single traffic flow.

## V. Traffic-Aware Relay Node Deployment: The General Case

We now address the general form of the deployment problem, i.e., the multi source multi traffic flow case.

### *A. Theoretical Solution in Continuous Space*

We first translate it into a graph equivalence. Define directed graph $G = (V, E)$, where $V = \{v_0, v_1, \ldots, v_M, v_{M+1}, \ldots\}$, $E = \{e_1, e_2, \ldots\}$. Let $v_i = s_i$ for $i = 0, 1, \ldots, M$. Here, vertices $v_j$, $j \geq M + 1$, are called *merge vertices* whose function will be explained later. Let $e_1, e_2, \ldots$ denote the edges that connect the vertices in $V$, where traffic flows can only pass an edge along its direction. The choice of $v_j$, $j \geq M+1$ and $e_i$ are to be determined later. Let $\lambda_{e_i}$ be the sum of average data rates of the traffic flows passing through edge $e_i$. Let $L_{e_i}$ be the length of the edge $e_i$, $n_{e_i}$ be the number of the R-nodes assigned on edge $e_i$ and $\mathbb{E}_{e_i}$ be the maximum energy consumption of an R-node on edge $e_i$.

As an example, Fig. 4 shows a simple case of two sources $s_1$ and $s_2$ with the base station $s_0$. By definition, we have $v_0 = s_0$, $v_1 = s_1$ and $v_2 = s_2$. Apparently, one deployment strategy is to place the R-nodes along $e_1$ and $e_2$, and the traffic flows can then be relayed to $s_0$ along these two edges separately. Alternatively, we can also find a *merge vertex* $v_3$ and deploy R-nodes along $e'_1$, $e'_2$ and $e'_3$; the traffic flows then can be relayed from $s_1$ and $s_2$ via $e'_1$ and $e'_2$, merged at $v_3$, and arrive at $s_0$ via $e'_3$. Surely there can be other relay node deployment schemes with different graph topology, but they all share one common feature that the network lifetime is bounded by the edge containing the R-node with the maximum energy cost. Note that each edge is directed from a start point to an end point, which is exactly the cases we have discussed in last section. Thus depending on whether one or multiple flows are relayed by an edge, we can apply Theorems 1 or 2 and have

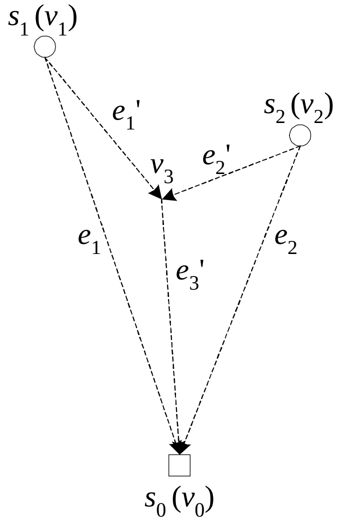


Fig. 4: An example on deployment for multi-source with multi-traffic flows.

$$\mathbb{E}_{e_i} = \mathbb{E}_{single}(L_{e_i}, n_{e_i}, \lambda_{e_i}) = \lambda_{e_i}[2c + (\frac{L_{e_i}}{n_{e_i}})^\alpha] \approx \lambda_{e_i}(\frac{L_{e_i}}{n_{e_i}})^\alpha\ .$$

Given that $\sum_{e_i \in E} n_i = N$, to achieve $\min\max_{e_i \in E}\ \ \mathbb{E}_{e_i}$, we need $\mathbb{E}_{e_1} = \mathbb{E}_{e_2} = \ldots$, which follows

$$\frac{\sqrt[\alpha]{\lambda_{e_1}}L_{e_1}}{n_{e_1}} = \frac{\sqrt[\alpha]{\lambda_{e_2}}L_{e_2}}{n_{e_2}} = \ldots = \frac{\sum_{e_i \in E}(\sqrt[\alpha]{\lambda_{e_i}}L_{e_i})}{\sum_{e_i \in E} n_i} \qquad (1)$$

The remaining task thus becomes finding the appropriate graph topology that achieves $\min \sum_{e_i \in E}(\sqrt[\alpha]{\lambda_{e_i}}L_{e_i})$. Once found, the number of R-nodes on each edge can be determined by Eq. (1) and the deployment then follows Theorem 1. We thus have the following observation:

*Observation 1:* The optimal solution to the general problem of multi source multi traffic flow is equivalent to minimizing the total weighted length of the edges that connect all the sources and the sink (allowing a set of merge vertices), where the weight on an edge $e_i$ is $\sqrt[\alpha]{\lambda_{e_i}}$.

The above problem is a generalized version of the Euclidian Steiner Minimum Tree problem, which is known NP-hard [26]. A heuristic is proposed in [25], which first constructs a graph

```
Algorithm RnodeAssignment()
1:  for $e_i \in E$, do $n_{e_i} \leftarrow 1$;
2:  $N \leftarrow N - |E|$;
3:  while $N > 0$, do
4:      find $e_i \in E$ such that $e_i$ has the largest energy
            cost $\mathbb{E}_{e_i} = \mathbb{E}_{single}(L_{e_i}, n_{e_i}, \lambda_{e_i})$;
5:      $n_{e_i} \leftarrow n_{e_i} + 1$;
6:      $N \leftarrow N - 1$;
7:  end while;
8:  return $n_{e_i}$ for all $e_i \in E$;
```

Fig. 5: The algorithm for discrete R-node assignment on edges.

topology by adding non-merge vertices one by one and then use a backtrack algorithm to optimize each size-$5$ component (a steiner-tree-like structure containing $5$ outer vertices and $3$ inner vertices) on the constructed graph topology.

In the construction, non-merge vertices can be added by two ordering schemes: 1) Min-Min ordering, where each added vertex minimizes the increased total weighted edge length (similar to the minimum spanning tree construction, but complicated due to creating a merge vertex at each step), and 2) Max-Min ordering, where each added vertex maximizes the minimum of the increased total weighted edge length. For each of the orderings an algorithm has been designed [25]. Unfortunately, no bounds were found for these two algorithms, and when $M$ increases over $10$, either one may return sub-optimal results.

Interestingly enough, our analysis shows that the sub-optimal results by different orderings are often stuck at different local optimums, even though they are designed to avoid being stuck too early before the size-$5$ component optimization stage. This motivates us to implement a hybrid algorithm that uses both orderings complementarily to bypass local optimums. Specifically, we start by adding non-merge vertices in one ordering, then switch to the other after $k$ vertices have been added, where $k$ is enumerated from $0$ to $M$. During our performance evaluation, we find that this hybrid algorithm successfully returns optimal results on all those test cases ($M \leq 15$) that can be verified within acceptable timeframes.

### B. Practical Solution on Discrete R-node Deployment

So far we have solved the problem of finding the graph topology, i.e., the location of the merge vertices, which minimize the maximum energy cost on an R-node. However, directly solving Eq. (1) may yield a fractional number of R-nodes being assigned to an edge. Our experience shows that a naive rounding to the closest integers can suffer from up to $40\%$ performance degradation. To build a practical solution, in this section, we develop algorithms for optimal discrete R-node assignment and merge vertices adjustments.

*1) Optimal Discrete R-node Assignment:* We develop a greedy algorithm (see Fig. 5) for the discrete R-node assignment problem, which assign each edge an integer number of R-nodes. It starts from the assignment with one R-node on each edge (line 1-2), which by Theorem 1, should be placed at the start point of each edge. Then we add other R-nodes one by one to the edge with the maximum energy consumption (line 4-7). This algorithm is optimal, as shown by the following:

```
Algorithm EnergyBalance()
$V_{min} \leftarrow V$;
$\mathbb{E}_{min} \leftarrow \max_{e_i \in E} \mathbb{E}_{e_i}$;
while true, do
    for each merge vertex $v \in V$, do
        adjust $v$ so as to balance $\mathbb{E}_{e_i}$ among all $e_i$
            connecting to $v$;
    end for
    if $\max_{e_i \in E} \mathbb{E}_{e_i} < \mathbb{E}_{min}$,
        $V_{min} \leftarrow V$;
        $\mathbb{E}_{min} \leftarrow \max_{e_i \in E} \mathbb{E}_{e_i}$;
    else break;
end while;
return $V_{min}$;
```

Fig. 6: The algorithm for balancing energy consumption on edges.

*Theorem 3:* Given the graph topology and any feasible R-node number, the RnodeAssignment() algorithm is optimal.

*Proof:* The proof is done by induction on the number of given R-nodes $N$. The key idea is: When an additional R-node is assigned to the current induction hypothesis, if an assignment better than the assignment achieved by our algorithm exists, then to improve our assignment, at least one R-node can be found to move from some other edge to the edge with the maximum energy cost in our assignment. Then without the additionally assigned R-node, the induction hypothesis can be further improved by moving the newly founded R-node and thus cause contradiction. The full proof can be found in [21]. ■

*2) Merge Vertex Adjustment:* Next we adjust the merge vertices to further balance the energy consumption among different edges. For example, if there is an edge that is short enough; then even deploying one R-node can lead to waste, i.e., when the network gets depleted, the residual energy of this R-node is still high. To this end, we develop two algorithms to balance the energy consumption on different edge and avoid such situations. Fig. 6 and Fig. 7 show the details.

EnergyBalance() (Fig. 6) proceeds iteratively (the **while** loop) to balance energy consumption among edges connecting to each merge vertex. In each iteration (the **for** loop), it tries to adjust the location of a merge vertex $v$ by solving equations

$$\lambda_{e_1}(\frac{\overline{v_1 v}}{n_{e_1}})^{\alpha} = \lambda_{e_2}(\frac{\overline{v_2 v}}{n_{e_2}})^{\alpha} = \lambda_{e_3}(\frac{\overline{v_3 v}}{n_{e_3}})^{\alpha} \ ,$$

where $e_1 = (v, v_1)$, $e_2 = (v, v_2)$ and $e_3 = (v, v_3)$. It is possible that $v$ has more than three edges connecting to it. In this case, we explore all 3-combinations that contain the edge with the maximum energy consumption, and use the solution that minimizes the maximum energy consumption among these edges. Note that $|E|$ is bounded by $(2 \times M - 1)$ [25], thus the computation complexity is polynomial and our

```
Algorithm AdjustMergeVertex()
1:  $n_{e_i}$, $e_i \in E \leftarrow$ RnodeAssignment();
2:  $V_{min} \leftarrow$ EnergyBalance();
3:  $E_{min} \leftarrow E$;
4:  $\mathbb{E}_{min} \leftarrow \max_{e_i \in E} \mathbb{E}_{e_i}$;
5:  while true, do
6:      $V_{temp} \leftarrow V_{min}$; $E_{temp} \leftarrow E_{min}$;
7:      $\mathbb{E}_{temp} \leftarrow \mathbb{E}_{min}$;
8:      for each merge vertex $v \in V_{min}$, do
9 :         $V \leftarrow V_{min}$; $E \leftarrow E_{min}$;
10:         combine $v$ with closest vertex for $V$ and $E$;
11:         $n_{e_i}$, $e_i \in E \leftarrow$ RnodeAssignment();
12:         $V \leftarrow$ EnergyBalance();
13:         if $\max_{e_i \in E} \mathbb{E}_{e_i} < \mathbb{E}_{temp}$,
14:             $V_{temp} \leftarrow V$; $E_{temp} \leftarrow E$;
15:             $\mathbb{E}_{temp} \leftarrow \max_{e_i \in E} \mathbb{E}_{e_i}$;
16:         end if
17:     end for
18:     if $\mathbb{E}_{temp} < \mathbb{E}_{min}$,
19:         $V_{min} \leftarrow V_{temp}$; $E_{min} \leftarrow E_{temp}$;
20:         $\mathbb{E}_{min} \leftarrow \mathbb{E}_{temp}$;
21:     else break;
22: end while;
23: return $V_{min}$ and $E_{min}$;
```

Fig. 7: The algorithm for merge vertex adjustment.

experience shows that the algorithm is fast in practice.

AdjustMergeVertex() (Fig. 7) takes the graph topology generated by the theoretical solution as an input. It first assigns R-nodes and does energy balancing (line 1-4). Then in each iteration (the **while** loop), it tries to combine each merge vertex with its closest vertex and keeps the combination that yields the largest reduction on the maximum energy cost among edges. Also during each try, it reassigns R-nodes and re-balance the energy consumption globally (line 11-12), so as to bypass local optimums. In next section, we will show that our solution, which considers both theoretical optimality and practical issues, has achieved excellent performance with good efficiency and balanced energy consumption.

## VI. Performance Evaluation

We evaluate our solution by both numerical analysis and $ns$-2 simulations. We adopt similar configurations from [9][27][16] in our evaluation. Specifically, we deploy $5$ to $25$ S-nodes by uniform distribution in a field of $5000m \times 5000m$ with the sink positioned at the center. The normalized data rate of each S-node is randomly picked from $(0, 1]$. For each number of S-nodes, $10$ topologies are generated. Each point in the figures thus represents the average with an error bar showing the standard deviation.

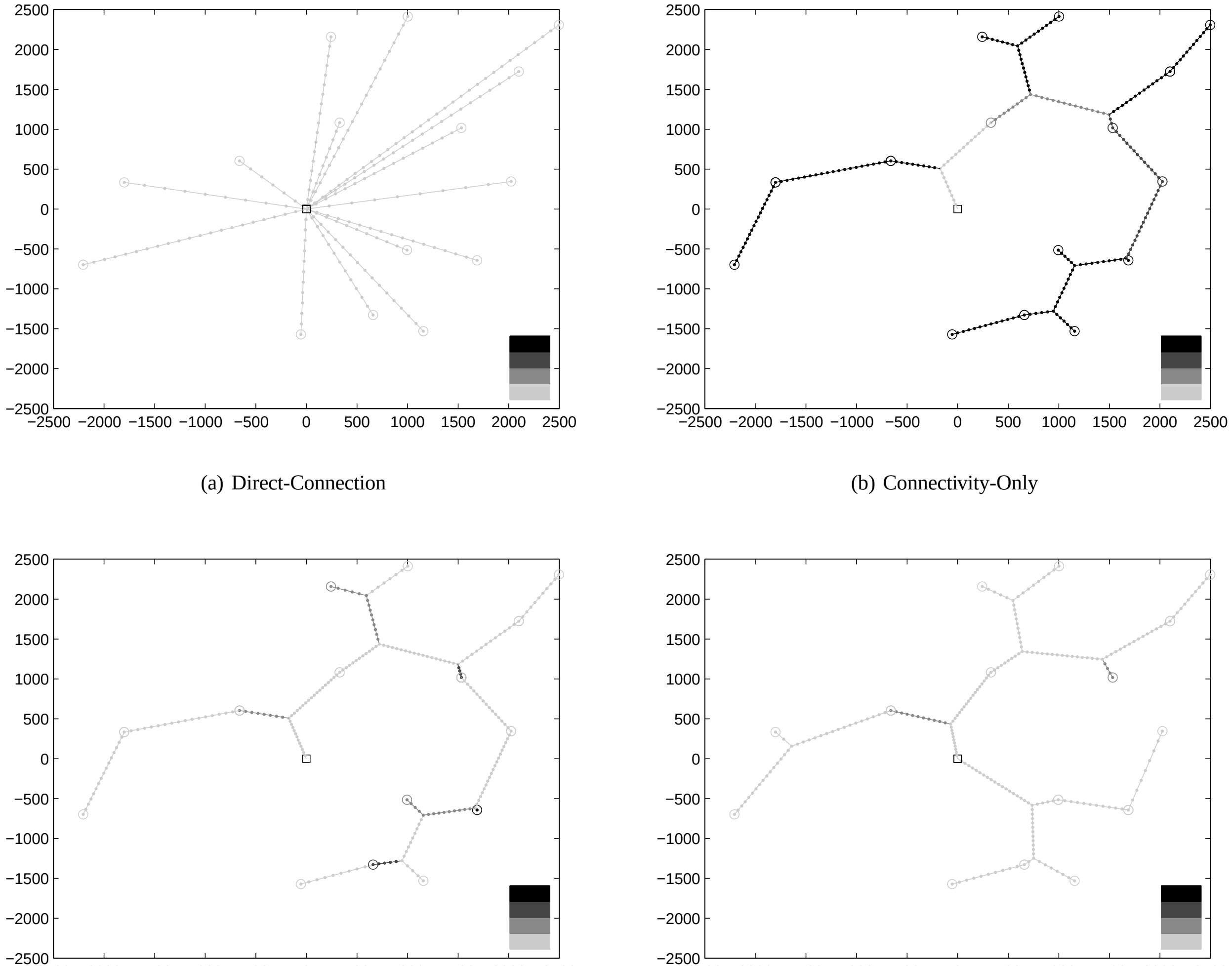
For comparison, we implemented three deployment approaches, namely *Direct-Connection*, *Connectivity-Only* and *Half-Traffic-Aware*. Direct-Connection connects each S-node with the sink by a dedicated data path (an edge) where R-nodes are deployed by our algorithm in Section V-B. It is the most straightforward approach and serves as a base-line. Connectivity-Only is chosen from a state-of-the-art scheme proposed in [27], which optimizes the system performance by considering connectivity only. There are multiple versions of the scheme. For better performance, we use the 1-connectivity version (i.e., there is at least one data path from each S-node to the sink) and further enhance it with a better approximation for Euclidean steiner minimum tree [25] (instead of minimum spanning tree) to construct the graph topology. The Half-Traffic-Aware approach uses the same graph topology as Connectivity-Only but assigns R-nodes by our algorithm proposed in Section V-B. It is used as a reference to help understand the impacts of the graph topology (by comparing with our solution) as well as the discrete R-node assignment algorithm (by comparing with Connectivity-Only). Fig. 8 illustrates how the three approaches and our solution deploy R-nodes by a test case of $15$ S-nodes used in our evaluation. Our solution is labeled by *Full-Traffic-Aware*.

Three metrics are used for evaluation. The first one is the *network lifetime*, defined as the lifetime of the first depleted R-node. In practice, this usually requests to dispatch a technician to replace the battery of this R-node. As sending a technician is costly, it is usually preferred that all the batteries are replaced. Thus, the first depleted node can serve as a good indicator for the end of the network lifetime. The second metric is the *residual energy*, defined as the residual energy of all R-nodes at the end of the network lifetime. Since all batteries are expected to be replaced at the same time, lower residual energy indicates less energy wastes on the removed batteries. The third metric is the *energy efficiency*, defined as the amount of traffic relayed to the sink by per unit energy cost. We consider this metric on purpose as we want to evaluate whether our solution extends the network lifetime at the expense of energy inefficiencies, as the phenomenon discussed in [17].

We set $\alpha = 4$ [17] and $R_{max} = 500m$. The initial energy for each node is set to $\mathbb{E} = T_{min} \cdot 10^8$, where $T_{min}$ is the minimum network lifetime requested by the application and is set to $1000$. With these application parameters ($\mathbb{E}$, $T_{min}$, $R_{max}$), for each test case (the number of S-nodes $M$ and their locations $s_i$ and data rates $\gamma_i$), the number of R-nodes $N$ is set to the minimum value computed by the following constraint (which is the number of R-nodes required by the base-line scheme Direct-Connection for a feasible deployment):

$$\begin{cases} \left(\sum_{i=1}^{M} \sqrt[\alpha]{\gamma_i} \cdot \overline{s_i s_0}\right) / \overline{N} \le \sqrt[\alpha]{\dfrac{\mathbb{E}}{T_{min}}} \\ n_i = \left\lceil \dfrac{\overline{N} \cdot \sqrt[\alpha]{\gamma_i} \cdot \overline{s_i s_0}}{\sum_{j=1}^{M} \sqrt[\alpha]{\gamma_j} \cdot \overline{s_i s_0}} \right\rceil \\ \max\limits_{i=1 \ldots M} \left( \dfrac{\overline{s_i s_0}}{n_i} \right) \le R_{max} \\ N = \sum\limits_{i=1}^{M} n_i \end{cases}$$

For ease of comparison, all results are normalized by the base-line scheme Direct-Connection.



(a) Direct-Connection

(b) Connectivity-Only

(c) Half-Traffic-Aware

(d) Full-Traffic-Aware

Fig. 8: An illustration of different deployment approaches and their residual energy distributions at the end of the network lifetime of each approach. The sink is denoted by the small square at the center. S-nodes are denoted by small circles. R-nodes are denoted by small diamond dots. Each approach uses the same number of R-nodes ($230$). Residual energy is demonstrated in grey scale, where darker color denotes higher residual energy. A grey scale reference is shown at the bottom right corner of each deployment.

### A. Numerical Results

Given a practical solution with the graph topology and the number of R-nodes on each edge, the network lifetime can be estimated as[1]

$$T = \min_{e_i \in E} \frac{\mathbb{E}}{\lambda_{e_i} \cdot (\frac{L_{e_i}}{n_{e_i}})^{\alpha}} \ .$$

The total residual energy is

$$\mathbb{E}_{residual} = \sum_{e_i \in E} \left(\mathbb{E} - \lambda_{e_i} \cdot (\frac{L_{e_i}}{n_{e_i}})^{\alpha} \cdot T\right) \cdot n_{e_i} \ .$$

And the energy efficiency is

$$\frac{T \cdot \sum_{i=1}^{M} \gamma_i}{N \cdot \mathbb{E} - \mathbb{E}_{residual}} = \frac{\sum_{i=1}^{M} \gamma_i}{\sum_{e_i \in E} \lambda_{e_i} \cdot (\frac{L_{e_i}}{n_{e_i}})^{\alpha} \cdot n_{e_i}} \ .$$

Fig. 9 shows the results of the network lifetime with different number of S-nodes. As the number of S-nodes increases, the lifetime of both Half- and Full-Traffic-Aware increase faster and is much higher than that of Direct-Connection and Connectivity-Only. With $25$ S-nodes, Half-Traffic-Aware performs over $11$ times of Direct-Connection and $5$ times of Connectivity-Only, while Full-Traffic-Aware further rises to $15$ times and $7$ times, respectively, which is $40\%$ higher than Half-Traffic-Aware. This demonstrates the importance of

[1] Following our analysis, we omit the small constant $c$ here. In our ns-2 simulation, all the practical factors (e.g., $c$) are included.

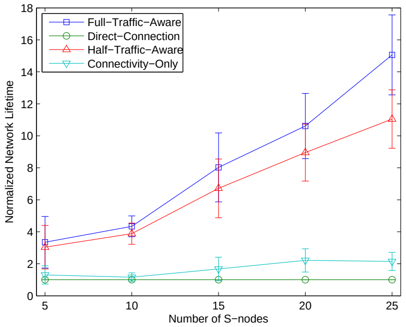


Fig. 9: Normalized network lifetime by numerical analysis.

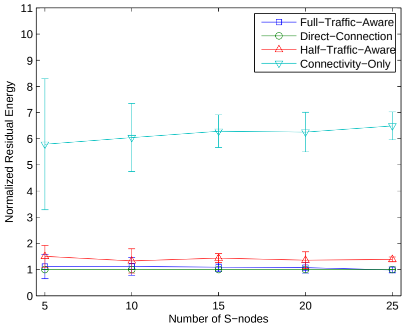


Fig. 10: Normalized residual energy by numerical analysis.

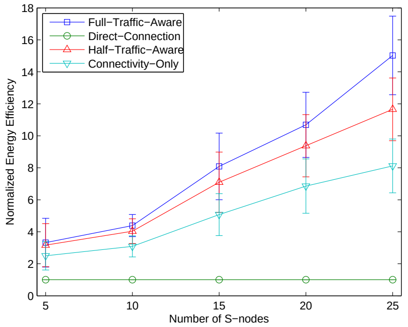


Fig. 11: Normalized energy efficiency by numerical analysis.

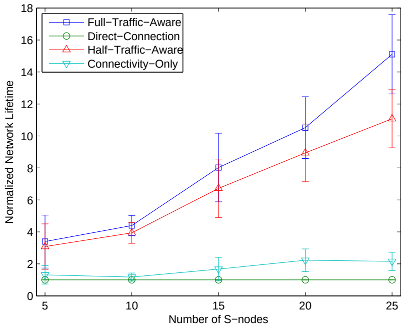


Fig. 12: Normalized network lifetime by $ns$-2 simulations.

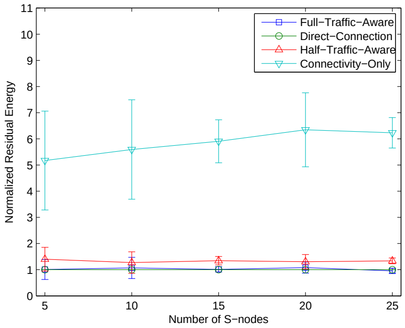


Fig. 13: Normalized residual energy by $ns$-2 simulations.

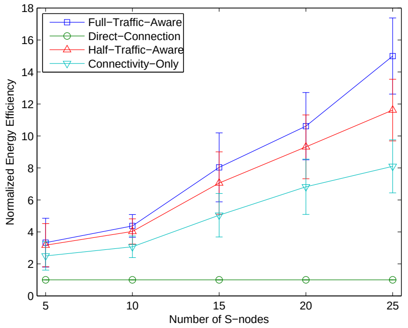


Fig. 14: Normalized energy efficiency by $ns$-2 simulations.

considering the traffic patterns during both graph topology selection (finding merge vertices) and node deployment stages (discrete R-node assignment and merge vertex adjustments).

Another interesting observation is that the lifetime of Connectivity-Only first rises and then drops slightly. A close investigation reveals the reason behind is that the energy hole phenomenon [17] becomes more significant when the number of S-nodes increases. Fig. 8 shows the residual energy distributions of four deployment strategies on a test case of $15$ S-nodes used in our evaluation. The energy hole problem can be clearly seen in Fig. 8b, where R-nodes close to the sink are depleted while most of other R-nodes still have more than $75\%$ of the energy. As the number of S-nodes increases, more traffic will accumulate close to the sink. This dramatically reduces the lifetime if the deployment is not aware of such traffic accumulations, e.g., the Connectivity-Only scheme. On the other hand, the other two schemes and our solution successfully avoid this problem by using algorithms that result in deploying more R-nodes close to the sink, as illustrated in Fig. 8a, Fig. 8c and Fig. 8d. In addition, there are still several edges with the residual energy more than $50\%$ of the initial energy in Fig. 8c. This is because Half-Traffic-Aware uses the same graph topology as Connectivity-Only, which is computed without traffic-awareness.

Fig. 10 shows the results of the total residual energy under different number of S-nodes (note the value is the lower the better). It is not surprising that the Direct-Connection, Half-Traffic-Aware and our Full-Traffic-Aware solution have much less total residual energy than Connectivity-Only, since the energy consumption of the former three schemes is more balanced by assigning more R-nodes to the edges with higher traffic volumes. However, as Half-Traffic-Aware uses traffic-blind graph topologies as Connectivity-Only, it runs the second higher. This also matches the residual energy distributions shown in Fig. 8, where Direct-Connection and our solution have more balanced distributions than Half-Traffic-Aware.

Fig. 11 shows the energy efficiencies of different deployment strategies under different number of S-nodes. It follows a similar trend to the network lifetime with one exception that Connectivity-Only has much better energy efficiency than Direct-Connection. This is because for the Connectivity-Only, most of R-nodes have not yet spent much energy when the first R-node dies. Nevertheless, our Full-Traffic-Aware solution still achieves the best energy efficiency and delivers about $15$ times of the traffic than Direct-Connection with the same mount of energy consumed, which shows that the extension of the network lifetime by our solution is not at the expense of energy inefficiencies.

### B. $ns$-2 Simulations

To further evaluate our solution, we conducted extensive simulations by $ns$-2, which consider both sending and receiving energy consumption, as well as wireless communication loss, collisions and other practical issues. A simple protocol is designed for data collection. The sink broadcasts a control message to start data collection. Each S-node then senses the environment at a predefined average rate and transmits the sensed data. Data losses are handled by both end-to-end and per-hop retransmissions. We modified the standard MAC layer to support dynamically adjusting transmission range by using different power. For consistency, we use the parameters adopted from [17] as in previous sections, i.e., $\alpha = 4$ and $c = 4500$ for both sending and receiving.

Fig. 12, Fig. 13 and Fig. 14 show the results of network lifetime, residual energy and energy efficiency under different number of S-nodes, respectively. It is easy to see that their trends match our numerical analysis well, which validates the correctness and effectiveness of our approach and analysis. By a careful comparison, we find that the simulation results in general are slightly better than those of the theoretical analysis, where the results of the total residual energy is the most obvious. A closer look reveals that by the communication range control, the wireless losses and collisions happen infrequently in all test cases. The only hot-spot identified is the area closer to the sink in the Direct-Connection scheme. Recall the example shown in Fig. 8a, where the R-nodes in this area are very close to each other and easy to cause collisions even under the communication range control. This slightly degrades the performance of Direct-Connection and also makes other schemes better after the normalization.

## VII. Conclusion and Future Work

In this paper, we presented an in-depth study on the traffic-aware relay node deployment problem. We developed optimal solutions for the case of one source node, both with single and multiple traffic flows. We showed however that the general problem is difficult, and existing connectivity-guaranteed solutions cannot be directly applied. We thus transformed our problem into a generalized version of the Euclidean Steiner Minimum Tree problem (ESMT) and proposed a hybrid algorithm. To further improve the performance, we also developed algorithms for discrete relay node assignment and further adjustments. We evaluated our solution by both numerical results and $ns$-2 simulations and observed an up to $14$ and $6$ times of improvement of the network lifetime than the Direct-Connection scheme and a state-of-the-art Connectivity-Only algorithm, respectively.

In the future, we would like to conduct some real experiments to evaluate our traffic-aware strategy. We also plan to consider more practical issues, especially deployment in 3-D space like a building, and deployment where R-nodes can only be placed within some feasible areas. Another direction is to consider survivability within our design to support fault-tolerance.

## References

[1] *Structural Health Monitoring for Guangzhou New TV Tower using Sensor Networks*. [Online]. Available: http://www.cse.polyu.edu.hk/benchmark/

[2] J. Bredin, E. Demaine, M. Hajiaghayi, and D. Rus, "Deploying Sensor Networks with Guaranteed Capacity and Fault Tolerance," in *ACM MobiHoc*, 2005.

[3] C. Buragohain, D. Agrawal, and S. Suri, "Power Aware Routing for Sensor Database," in *IEEE INFOCOM*, 2005.

[4] J. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," in *IEEE INFOCOM*, 2000.

[5] C. Gui and P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks," in *ACM MobiCom*, 2004.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *IEEE HICSS*, 2000.

[7] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "Prolonging Sensor Network Lifetime with Energy Provisioning and Relay Node Placement," in *IEEE SECON*, 2005.

[8] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2005.

[9] A. Kashyap, S. Khuller, and M. Shayman, "Relay Placement for Higher Order Connectivity in Wireless Sensor Networks," in *IEEE INFOCOM*, 2006.

[10] J. Ko, Y. Ni, H. Zhou, J. Wang, and X. Zhou, "Investigation Concerning Structural Health Monitoring of an Instrumented Cable-Stayed Bridge," *Structure and Infrastructure Engineering*, 2008.

[11] N. Li and J. Hou, "Localized Topology Control Algorithms for Heterogeneous Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 6, pp. 1313–1324, December 2005.

[12] N. Li, J. Hou, and L. Sha, "Design and Analysis of an MST-Based Topology Control Algorithm," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, May 2005.

[13] Q. Li, J. Aslam, and D. Rus, "Online Power-Aware Routing in Wireless Ad-hoc Networks," in *ACM MobiCom*, 2001.

[14] G. Lin and G. Xue, "Steiner Tree Problem with Minimum Number of Steiner Points and Bounded Edge-Length," *Information Processing Letters*, vol. 69, pp. 53–57, 1999.

[15] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," in *IEEE INFOCOM*, 2001.

[16] S. Mistra, S. D. Hong, G. Xue, and J. Tang, "Constrained Relay Node Placement in Wireless Sensor Networks to Meet Connectivity and Survivability Requirements," in *IEEE INFOCOM*, 2008.

[17] S. Olariu and I. Stojmenovic, "Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting," in *IEEE INFOCOM*, 2006.

[18] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transit Power Adjustment," in *IEEE INFOCOM*, 2000.

[19] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young, and J. Porter, "LUSTER: Wireless Sensor Network for Environmental Research," in *ACM SenSys*, 2007.

[20] S. Singh, M. Woo, and C. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," in *ACM MobiCom*, 1998.

[21] F. Wang, D. Wang, and J. Liu, "Traffic-Aware Deployment for Data Collections using Wireless Sensor Networks," Simon Fraser University, Tech. Rep., 2008.

[22] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks," in *IEEE INFOCOM*, 2001.

[23] G. WernerAllen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and Yield in a Volcano Monitoring Sensor Network," in *USENIX OSDI*, 2006.

[24] K. Xu, H. Hassanein, and G. Takahara, "Relay Node Deployment Strategies in Heterogeneous Wireless Sensor Networks: Multiple-Hop Communication Case," in *IEEE SECON*, 2005.

[25] G. Xue, T. P. Lillys, and D. E. Dougherty, "Computing the Minimum Cost Pipe Network Interconnecting One Sink and Many Sources," *SIAM Journal of Optimization*, vol. 10, no. 1, pp. 22–42, October 1999.

[26] G. Xue and Y. Ye, "An Efficient Algorithm for Minimizing a Sum of Euclidean Norms with Applications," *SIAM Journal of Optimization*, vol. 7, no. 4, pp. 1017–1036, November 1997.

[27] W. Zhang, G. Xue, and S. Misra, "Fault-tolrant Relay Node Placement in Wireless Sensor Networks: Problems and Algorithms," in *IEEE INFOCOM*, 2007.