# Traffic-Aware Relay Node Deployment: Maximizing Lifetime for Data Collection Wireless Sensor Networks

Feng Wang, *Student Member*, *IEEE*, Dan Wang, *Member*, *IEEE*, and Jiangchuan Liu, *Senior Member*, *IEEE* 

Abstract—Wireless sensor networks have been widely used for ambient data collection in diverse environments. While in many such networks the 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 only connectivityguaranteed 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 for all test cases where the continuous space optimal solution can be computed within acceptable time frames, the network lifetime achieved by our solution is very close to the upper bound of the optimal solution (the difference is less than 13.5 percent). Moreover, it achieves up to 6-14 times improvement over the existing traffic-oblivious strategies.

Index Terms—Wireless sensor networks, data collection, deployment, traffic-aware, relay node.

## **1** 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 [16] 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 [19], [22], [11], [12] and routing design [21], [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

Recommended for acceptance by S. Olariu.

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 100 Hz. 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], [17], most of which however focused on maintaining network connectivity. Given the heterogenous traffic flows

<sup>•</sup> F. Wang and J. Liu are with the School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, British Columbia, Canada V5A 1S6. Email: {fwa1, jcliu}@cs.sfu.ca.

<sup>•</sup> D. Wang is with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. Email: csdwang@comp.polyu.edu.hk.

Manuscript received 28 May 2009; revised 30 Jan. 2010; accepted 22 July 2010; published online 3 Jan. 2011.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2009-05-0243. Digital Object Identifier no. 10.1109/TPDS.2011.20.



Fig. 1. An example of relay node deployment: (a) connectivity-only scheme; (b) traffic-aware scheme.  $s_1$ ,  $s_2$  are sources with data rate of 0.6 and 0.3.  $s_0$  is the sink. Given N relay nodes, by scheme (a) which only considers connectivity, nodes relaying the traffic from v to  $s_0$  will die much earlier than those relaying from  $s_1$  and  $s_2$  to v, while by strategically deploying more nodes on section  $(v, s_0)$  (from less busy section  $(s_2, v)$ ), the network lifetime is prolonged.

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. 1a maximizes the network lifetime, i.e., each of the  $\frac{1}{3}$  of the relay nodes is deployed on the sections of  $(s_1, v)$ ,  $(s_2, v)$ , and  $(v, s_0)$ . We can see, however, given the traffic pattern, the deployment scheme that moves some relay nodes from less traffic intensive section  $(s_2, v)$  to heavily loaded  $(v, s_0)$  will achieve better performance, as shown in Fig. 1b.

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 euclidean Steiner Minimum Tree (ESMT) problem and develop a hybrid algorithm that successfully returns optimal results with all test cases that can be verified within acceptable time frames. 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 for all test cases where the continuous space optimal solution can be computed within acceptable time frames, the network lifetime achieved by our solution is very close to the upper bound of the optimal solution (the difference is less than 13.5 percent). Moreover, 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 six times than the state-of-the-art algorithm considering connectivity only. The remaining part of the paper proceeds as follows: Section 2 presents the related work. We outline the system models and the problem description in Section 3. Section 4 proposes solutions to several case studies which can be used as building blocks for the general problem. In Section 5, 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 6. Finally, Section 7 concludes our paper and gives directions of future work.

## 2 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 [19], 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 [21]. 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 assumed that the network node deployment is given, which often follows a random distribution.

Relay node deployment for WSNs has been studied in various contexts [27], [17], [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  $k \ge 2$ . Later [17] 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] focused on massive random relay node deployment and [7] stressed 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.

## **3 PROBLEM STATEMENT**

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. In addition, when aggregated together, these data

rates would not exceed the wireless communication capacity, i.e., there is no bottleneck in the network.

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  $S = \{s_1, s_2, \ldots, s_M\}$  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  $p_i = x_0x_1 \ldots x_{l_i}$  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 is defined as the lifetime of the first depleted relay node. The problem thus can be formulated as follows:

**Traffic-Aware Deployment Problem**: Given N, the total number of R-nodes to be deployed, where  $N \ge M$ , finds 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, given that all the relay nodes have the same residual energy initially, the network lifetime is critically bounded by the nodes with the highest energy costs. Thus, we are interested in minimizing the maximum energy consumption among the R-nodes, i.e.,

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

Note that the summation here indicates that an R-node can undertake combined traffic flows of multiple sources if it is chosen in these paths. For ease of exposition, we summarize the notations used here and throughout the paper in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/ 10.1109/TPDS.2011.20. Denote  $R_{max}$  is the maximum communication range of an R-node. The deployment should satisfy the following constraints:

1. Communication range,

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

2. Forwarding path connectivity,

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

3. S-nodes and sink connectivity,

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

To simplify exposition, we associate each S-node with an Rnode at the same location (as shown in Constraint 3), which guarantees S-nodes are only involved in local short range communications with marginal costs 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 [18] will be used in this paper:

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



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

which can also be normalized as

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

where  $\alpha$  is generally greater than 1 with typical values between 2 and 6, and *c* is a small constant comparing with  $r^{\alpha}$ . The energy consumption for packet receiving is given by<sup>1</sup>

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

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], [20], [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.

## 4 TRAFFIC-AWARE R-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.

#### 4.1 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  $\frac{L}{N} \leq R_{max}$  for a feasible solution. Let the distance between the *i*th R-node and its next R-node/sink be  $r_i$ , i = 1, ..., 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, ..., 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:

**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  $\mathbb{E}_{single}(L, N, \gamma) = \gamma [2c + (\frac{L}{N})^{\alpha}]$ .

<sup>1.</sup> In the communication model, c is a small constant to represent the basic costs when the transceiver circuit works. Since receiving does not include the extra costs of generating signals that travel distantly,  $\mathbb{E}_{recv}$  is thus set to c.



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

**Proof.** A detailed proof can be found in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS. 2011.20. □

#### 4.2 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* Rnodes 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 section, it is easy to see that the R-nodes should be assigned such that

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

Consequently,

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

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

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

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

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

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

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

Clearly, we have

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



Fig. 4. An example for multisource with multitraffic flows.

which shows that merging these two flows leads to the minimum energy cost on an R-node. This result can be easily generalized 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.
- **Proof.** A detailed proof can be found in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS. 2011.20.

## 5 TRAFFIC-AWARE R-NODE DEPLOYMENT: THE GENERAL CASE

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

## 5.1 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  $v_j$ ,  $j \ge M + 1$ , 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  $v_j$ ,  $j \ge$ M + 1 and  $e_i$  is 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 merge vertices being placed at different locations or using different graph topologies, but the network lifetime of each scheme is always 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

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

To achieve  $\min \max_{e_i \in E} \mathbb{E}_{e_i}$ , we need  $\mathbb{E}_{e_1} = \mathbb{E}_{e_2} = \cdots$ , which follows

$$\frac{\sqrt[\alpha]{\lambda_{e_1}}L_{e_1}}{n_{e_1}} = \frac{\sqrt[\alpha]{\lambda_{e_2}}L_{e_2}}{n_{e_2}} = \dots = \frac{\sum_{e_i \in E} \left(\sqrt[\alpha]{\lambda_{e_i}}L_{e_i}\right)}{\sum_{e_i \in E} n_i}.$$
 (1)

Given that  $\sum_{e_i \in E} n_i = N$ , 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 edge directions and data rates can be easily determined.<sup>2</sup> The R-node number on each edge can be computed by (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  $\sqrt[6]{\lambda_{e_i}}$ .

The above problem is a generalized version of the Euclidean Steiner Minimum Tree problem,<sup>3</sup> which is NPhard [26]. A heuristic is proposed in [25], which first constructs a graph topology by adding non-merge vertices one by one and then use a backtrack algorithm to optimize each size-5 component on the constructed graph topology. In the construction, non-merge vertices can be added by two ordering schemes: 1) Min-Min, 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, 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 suboptimal results.

Interestingly enough, our analysis shows that the suboptimal 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 nonmerge vertices in one ordering, then switch to the other after *k* vertices has been added, where *k* is enumerated from 1 to M - 1. Appendix C, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety. org/10.1109/TPDS.2011.20, shows the details of the hybrid algorithm. During our performance evaluation, we find that

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

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

this hybrid algorithm successfully returns optimal results on all those test cases ( $M \le 15$ ) that can be verified within acceptable time frames.

## 5.2 Practical Solution on Discrete Node Deployment

So far, we have a solution for finding the graph topology, i.e., the location of the merge vertices, which minimizes the maximum energy cost on an R-node. However, directly solving (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 percent performance degradation. To build a practical solution, in this section, we develop algorithms for optimal discrete R-node assignment and merge vertices adjustments.

#### 5.2.1 Optimal Discrete R-Node Assignment

We develop a greedy algorithm (see Fig. 5) for the discrete R-node assignment problem, which assigns each edge an integer number of R-nodes. It starts from the assignment with one R-node on each edge (line 1), 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 (lines 3-4). This algorithm is optimal, as shown by the following:

- **Theorem 3.** Given the graph topology and any feasible R-node number, the RnodeAssignment() algorithm generates the optimal R-node assignment to the edges of the given graph topology such that the maximum energy costs among the edges are minimized.
- Proof. A detailed proof can be found in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS. 2011.20.

#### 5.2.2 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. We omit their details here due to space limitation. A full description of these two algorithms can be found in Appendix D, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/ 10.1109/TPDS.2011.20.

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.

<sup>2.</sup> A detailed discussion on deriving the optimal edge directions and data rates of a given graph toplogy can be found in [25].

<sup>3.</sup> Note that although the edge weights are determined by the interconnection of the graph topology, the graph topology and the edge lengths are, in turn, determined by the locations of the merge points, which may be chosen from anywhere within the sensing field. In particular, our problem is exactly like the euclidean Steiner tree problem, where the locations of the Steiner points have to be first chosen from the continuous plane, and then, the edge weights (i.e., the length of the edge in the case of the euclidean Steiner tree problem) can be determined. It is different from the Steiner tree problem, where all possible discrete locations of Steiner points have been given and the problem is to determine the interconnection network/tree topology that may use some of these given discrete locations.



Fig. 6. 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 different color scale, where redder and darker color denotes higher residual energy while greener and lighter color denotes lower residual energy. A color scale reference is shown at the bottom right corner of each deployment. (a) Direct-Connection. (b) Connectivity-Only. (c) Half-Traffic-Aware. (d) Full-Traffic-Aware.

## 6 **PERFORMANCE EVALUATION**

We evaluate our solution by both numerical analysis and *ns*-2 simulations. In this section, we present the numerical analysis and leave the *ns*-2 simulations in Appendix F, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.20. We adopt similar configurations from [9], [27], [17] in our evaluation. Specifically, we deploy 5-25 S-nodes by uniform distribution in a field of 5,000 m × 5,000 m 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 other approaches, namely, Direct-Connection, Connectivity-Only, and Half-Traffic-Aware. Direct-Connection connects each Snode with the sink by a dedicated data path (an edge) where R-nodes are deployed by our algorithm in Section 5.2. It is the most straightforward approach and serves as a baseline. Connectivity-Only is chosen from a series of stateof-the-art schemes proposed in [15], [27], which optimize the system performance by considering connectivity only. For better performance, we use the one-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 5.2. 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. 6 illustrates how the three approaches and our solution deploy R-nodes by a test case of 15 S-nodes 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 [18].

We set  $\alpha = 4$  [18] and  $R_{max} = 500$  m. 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 1,000. The formulas used for our numerical analysis can be found in Appendix E, which can be found on the Computer Society Digital Library at http://doi. ieeecomputersociety.org/10.1109/TPDS.2011.20. In the following sections, we investigate the impacts of different number of R-nodes and S-nodes, respectively.

#### 6.1 Impact of R-Node Number Selection

We first set the number of S-node to 25 and compute the numerical results to analyze how the performance of different solutions changes with the number of R-nodes. Given that the field is  $5,000 \text{ m} \times 5,000 \text{ m}$  with  $R_{max} = 500 \text{ m}$ , for the Direct-Connection scheme, the average of the minimum integer number of R-nodes required to work properly (i.e., at least the basic connectivity is guaranteed) is 100 with the upper bound of the minimum integer number as high as 200 (i.e., averagely around four R-nodes and maximally eight R-nodes for one S-node). As will be discussed later that Direct-Connection always needs a higher minimum number of R-nodes than other schemes, the numerical analysis is thus set to start from 200 R-nodes, with the range from 200 to 500.

Fig. 7 shows the results of the network lifetime with different number of R-nodes. Interestingly enough, when the



Fig. 7. Normalized network lifetime with different number of R-nodes by numerical analysis.



Fig. 8. Normalized network lifetime with the number of R-nodes  $\leq 200$  by numerical analysis.

number of R-nodes is equal to or greater than 250, the trends of all solutions keep quite steady and are not very sensitive to the changes of the R-node number (note that the results are normalized by the Direct-Connection scheme, which flattens the slope of each solution). On the other hand, when the number of R-nodes is less than 250, the performance of Full-Traffic-Aware and Half-Traffic-Aware seems to drop a little. To investigate how these two schemes perform when the number of R-nodes is more comparable to the number of S-nodes, we temporarily relax the communication range constraint for Direct-Connection and further conduct numerical analysis with the number of R-nodes ranged from 80 to 200. The results are shown in Fig. 8. It can be seen that there are some small fluctuations in the figure. This is because as the number of R-nodes decreases, the marginal effects of the random topology variations and that only an integer number of R-nodes can be used on each edge of a graph topology may become more observable. Even though, the trends of Full-Traffic-Aware and Half-Traffic-Aware are still relatively stable with the performance only slightly decreased. Also, in Figs. 7 and 8, while Half-Traffic-Aware performs roughly up to 11 times of Direct-Connection and five times of Connectivity-Only, Full-Traffic-Aware further raises the gain up to 15 times and 7 times, respectively, which is 40 percent higher than Half-Traffic-Aware. This demonstrates the importance of considering the traffic patterns during both graph topology selection (finding merge vertices) and node deployment stages (discrete Rnode assignment and merge vertex adjustments).

Fig. 9 shows the results of the total residual energy with different number of R-nodes (note that 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. This also explains that the residual energy of Connectivity-Only



Fig. 9. Normalized residual energy with different number of R-nodes by numerical analysis.



Fig. 10. Normalized energy efficiency with different number of R-nodes by numerical analysis.

increases much faster than the other three solutions. In addition, as Half-Traffic-Aware uses traffic-blind graph topologies as Connectivity-Only, it runs the second higher. This also matches the residual energy distributions illustrated in Fig. 6, where Direct-Connection and our solution have more balanced distributions than Half-Traffic-Aware.

Fig. 10 shows the energy efficiencies of different deployment strategies with different number of R-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 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.

It is worth noting that while selecting different R-node number has only marginal impacts on the performance comparison, there does exist a minimum requirement on Rnode number to guarantee the WSN system working well. The required minimum R-node number varies with the given S-node number, their locations and data rates, and the used deployment strategy. During our numerical analysis, we find that the Direct-Connection scheme always needs a higher minimum R-node number than the other three schemes. We thus derive an upper bound on the minimum R-node number required by Direct-Connection for a given S-node number and their locations and data rates (see Appendix E, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/ 10.1109/TPDS.2011.20). We compute this bound for each test case and use the results as the default R-node number in the remaining of this section.

#### 6.2 Scalability with S-Node Number

We next investigate how our solution performs with different number of S-nodes. We also compute the results of the theoretical solution by (1) proposed in Section 5.1, which serve as a bound to evaluate our practical solution on discrete R-node deployment. It is also worth noting that for all test cases ( $M \le 15$ ) that can be verified within acceptable time frames, our hybrid algorithm successfully returns optimal graph topologies. In these cases, the theoretical solution actually serves as the upper bound of the *optimal solution*.

Fig. 11 shows the results of the network lifetime with different number of S-nodes. It is clear to see that our Full-Traffic-Aware scheme is very close to the theoretical solution with the difference less than 13.5 percent, and it



Fig. 11. Normalized network lifetime with different number of S-nodes by numerical analysis.

performs much better than the other solutions. 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. One 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 [18] becomes more significant when the number of S-nodes increases. Fig. 6 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. 6b, where R-nodes close to the sink are depleted while most of other R-nodes still have more than 75 percent 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 Figs. 6a, 6c, and 6d. In addition, there are still several edges with the residual energy more than 50 percent of the initial energy in Fig. 6c. This is because Half-Traffic-Aware uses the same graph topology as Connectivity-Only, which is computed without traffic-awareness.

Fig. 12 shows the results of the total residual energy with different number of S-nodes. As the number of S-nodes increases, all the five schemes follow a similar trend and are not very sensitive to the S-node number. As expected, the theoretical solution has the lowest residual energy, which is followed by Direct-Connection and our Full-Traffic-Aware solution. Half-Traffic-Aware runs the forth due to its traffic-blind graph topology selection and Connectivity-Only performs even worse due to lacking of traffic-awareness in both graph topology selection and node deployment stages.

Fig. 13 shows the energy efficiencies of different deployment strategies with different number of S-nodes. As in the analysis on the impact of different R-node number, the



Fig. 12. Normalized residual energy with different number of S-nodes by numerical analysis.



Fig. 13. Normalized energy efficiency with different number of S-nodes by numerical analysis.

energy efficiency also follows a similar trend to the network lifetime with the exception that Connectivity-Only has much better energy efficiency than Direct-Connection. Besides, our Full-Traffic-Aware solution performs almost the same as the theoretical solution.

## 7 CONCLUSION AND FUTURE WORK

In this paper, we presented an in-depth study on the trafficaware 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 connectivityguaranteed-only solutions cannot be directly applied here. We then transformed our problem into a generalized version of the ESMT problem 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 that for all test cases where the continuous space optimal solution can be computed within acceptable time frames, our solution is very close to the upper bound of the optimal solution. Moreover, our solution has 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.

Next, 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 3D space like a building, and deployment where R-nodes can only be placed within some feasible areas. We are also interested in investigating solutions of exploiting multiple base stations when the total traffic amount exceeds the capacity of one base station. Yet, another direction is to consider survivability within our design to support fault-tolerance.

#### ACKNOWLEDGMENTS

Feng Wang and Jiangchuan Liu's work is supported by a Canada NSERC Strategic Grant, an NSERC Discovery Grant, an NSERC DAS grant, and an MITACS Project Grant. Dan Wang's work is supported by grant Hong Kong PolyU/GY-G78, A-PB0R, A-PJ19, 1-ZV5W, and RGC/GRF PolyU 5305/08E. A preliminary version of this paper appeared in IEEE SECON '09.

#### REFERENCES

 Structural Health Monitoring for Guangzhou New TV Tower Using Sensor Networks, http://www.cse.polyu.edu.hk/ benchmark/, 2011.

- [2] J. Bredin, E. Demaine, M. Hajiaghayi, and D. Rus, "Deploying Sensor Networks with Guaranteed Capacity and Fault Tolerance," *Proc. ACM MobiHoc*, 2005.
- [3] C. Buragohain, D. Agrawal, and S. Suri, "Power Aware Routing for Sensor Database," *Proc. IEEE INFOCOM*, 2005.
- [4] J. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-Hoc Networks," Proc. IEEE INFOCOM, 2000.
- [5] C. Gui and P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks," Proc. ACM MobiCom, 2004.
- [6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," Proc. IEEE Hawaii Int'l Conf. System Sciences (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," Proc. IEEE Second Ann. Comm. Soc. Conf. Sensor and Ad Hoc Comm. and Networks (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," *Proc. 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," *Proc. Structure and Infrastructure Eng.*, 2008.
- [11] N. Li and J. Hou, "Localized Topology Control Algorithms for Heterogeneous Wireless Networks," *IEEE/ACM Trans. Networking*, vol. 13, no. 6, pp. 1313-1324, Dec. 2005.
- [12] N. Li, J. Hou, and L. Sha, "Design and Analysis of an MST-Based Topology Control Algorithm," *IEEE Trans. Wireless Comm.*, 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," Proc. 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] E.L. Lloyd and G. Xue, "Relay Node Placement in Wireless Sensor Networks," *IEEE Trans. Computers*, vol. 56, no. 1, pp. 134-138, Jan. 2007.
- [16] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava, "Coverage Problems in Wireless Ad-Hoc Sensor Networks," *Proc. IEEE INFOCOM*, 2001.
- [17] S. Misra, S.D. Hong, G. Xue, and J. Tang, "Constrained Relay Node Placement in Wireless Sensor Networks to Meet Connectivity and Survivability Requirements," *Proc. IEEE INFOCOM*, 2008.
- [18] S. Olariu and I. Stojmenovic, "Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting," *Proc. IEEE INFOCOM*, 2006.
- [19] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transit Power Adjustment," *Proc. IEEE INFOCOM*, 2000.
- [20] 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," Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys), 2007.
- [21] S. Singh, M. Woo, and C. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," Proc. ACM MobiCom, 1998.
- [22] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM*, 2001.
- [23] G. WernerAllen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and Yield in a Volcano Monitoring Sensor Network," *Proc. USENIX Symp. Operating Systems Design and Implementation* (OSDI), 2006.
- [24] K. Xu, H. Hassanein, and G. Takahara, "Relay Node Deployment Strategies in Heterogeneous Wireless Sensor Networks: Multiple-Hop Communication Case," Proc. IEEE Second Ann. Comm. Soc. Conf. Sensor and Ad Hoc Comm. and Networks (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 J. Optimization*, vol. 10, no. 1, pp. 22-42, Oct. 1999.
- [26] G. Xue and Y. Ye, "An Efficient Algorithm for Minimizing a Sum of Euclidean Norms with Applications," SIAM J. Optimization, vol. 7, no. 4, pp. 1017-1036, Nov. 1997.

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



Feng Wang (S'07) received the bachelor's and master's degrees in computer science and technology from the Tsinghua University, Beijing, China, in 2002 and 2005, respectively. He is currently working toward the PhD degree in the School of Computing Science at the Simon Fraser University, conducting researches on wireless sensor networks and peer-to-peer live streaming under the supervision of Prof. Jiangchuan Liu. His research interests include wireless

sensor networks, peer-to-peer live streaming, and distributed computing. He has been a student member of the IEEE and the IEEE Communications Society since 2007. In 2006 summer, he interned in the Wireless and Networking Group at Microsoft Research Asia, where he conducted research on peer-to-peer live video streaming. He also conducted research in the Department of Computing at the Hong Kong Polytechnic University as a visiting PhD student in 2008 spring, 2009 spring and fall, where his research was on data collections in wireless sensor networks. In 2010 spring, he visited the Wireless Ad Hoc Networks Lab in the Institute of Software at the Chinese Academy of Sciences, where he worked on the testbed and information system design for wireless sensor networks. He was awarded the Tsinghua University Scholarship for Excellent Student in 1998, 2000, and 2001. At the Simon Fraser University, he was awarded the SFU-CS Graduate Entrance Scholarship in 2005 and the Graduate Fellowship in 2007, 2008, and 2009. He was also awarded the Chinese Government Scholarship for Outstanding Selffinanced Students Studying Abroad in 2009.



**Dan Wang** (S'05-M'07) received the BSc degree from the Peking University, Beijing, China, in 2000, the MSc degree from the Case Western Reserve University, Cleveland, Ohio, in 2004, and the PhD degree from the Simon Fraser University, Burnaby, British Columbia, Canada, in 2007; all in computer science. He is currently an assistant professor in the Department of Computing, The Hong Kong Polytechnic University. His research interests include wireless

sensor networks, internet routing, and peer-to-peer networks. He is a member of the IEEE.



Jiangchuan Liu (S'01-M'03-SM '08) received the BEng degree (cum laude) from the Tsinghua University, Beijing, China, in 1999, and the PhD degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He is a recipient of Microsoft Research Fellowship (2000), Hong Kong Young Scientist Award (2003), and Canada NSERC DAS Award (2009). He is a corecipient of the Best Student Paper Award of IWQoS '08, the

Best Paper Award (2009) of IEEE ComSoc Multimedia Communications Technical Committee, and Canada BCNet Broadband Challenge Winner Award 2009. He is currently an associate professor in the School of Computing Science, Simon Fraser University, British Columbia, Canada and was an assistant professor in the Department of Computer Science and Engineering at The Chinese University of Hong Kong from 2003 to 2004. His research interests include multimedia systems and networks, wireless ad hoc and sensor networks, and peer-to-peer and overlay networks. He is a senior member of the IEEE and a member of Sigma Xi. He is an associate editor of the *IEEE Transactions on Multimedia*, an editor of the *IEEE Communications Surveys and Tutorials*, and an area editor of the *Computer Communications*. He is TPC vice chair for Information Systems of IEEE INFOCOM '11.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.