

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 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 for all test cases where the continuous space optimal solution can be computed within acceptable timeframes, 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%). Moreover, it achieves up to 6 to 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 Hong Kong, which is equipped with a large number of accelerom-

eters, 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][17], most of which however focused on maintaining network connectivity. Given the heterogenous traffic flows and the many-

- Feng Wang and Jiangchuan Liu are with the School of Computing Science, Simon Fraser University, British Columbia, V5A 1S6 Canada.
Email: {fwa1,jcliu}@cs.sfu.ca
- Dan Wang is with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.
Email: csdwang@comp.polyu.edu.hk

A preliminary version of this paper appeared in IEEE SECON'09.

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, 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. 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 for all test cases where the continuous space optimal solution can be computed within acceptable timeframes, 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%). 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 6 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

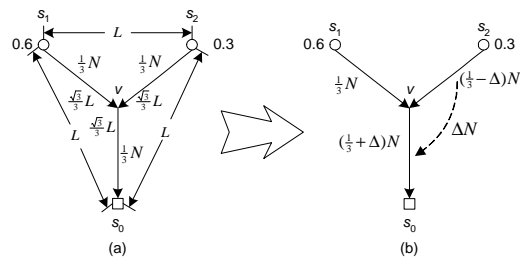


Fig. 1: An example of relay node deployment: (a) connectivity-only scheme; (b) traffic-aware scheme. s_1, s_2 are sources with data rate of 0.6 and 0.3. s_0 is the sink. Given N relay nodes, by scheme (a) which only considers connectivity, nodes relaying the traffic from v to s_0 will die much earlier 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.

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 $k \geq 2$. 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] 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.

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.

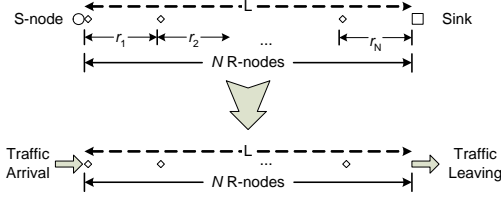


Fig. 2: An illustration of deployment for single source single flow and its generalization.

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 $S = \{s_1, s_2, \dots, s_M\}$ 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 γ_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 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 $N \geq M$, find the geographical locations for R-nodes $F = \{f_1, f_2, \dots, f_N\}$, together with their respective communication ranges $R = \{r_1, r_2, \dots, r_N\}$ and traffic paths for S-nodes $P = \{p_1, p_2, \dots, p_M\}$, 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.,

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

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. For ease of exposition, we summarize the notations used here and throughout the paper in Appendix A. 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,

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

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 short range communications with marginal costs and the network lifetime

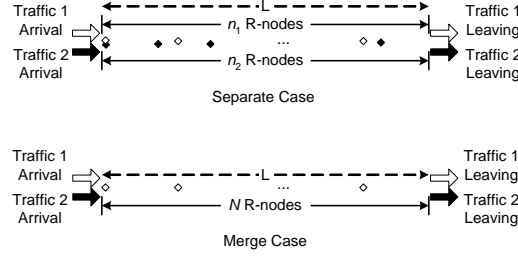


Fig. 3: An illustration of two deployment schemes for single source two traffic flows.

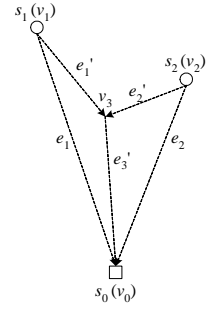


Fig. 4: An example for multi-source with multi-traffic flows.

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:

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

which can also be normalized as

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

where α is generally greater than 1 with typical values between 2 and 6, and c is a small constant comparing with r^α . The energy consumption for packet receiving is given by¹

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

Finally, it is worth noting that our network model can be easily extended to a hierarchical 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 γ_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 , $i = 1, \dots, N$, the energy cost for the i -th R-node is

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

1. 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 .

Since $\sum_{i=1}^N r_i = L$, it is easy to see that the solution to $\min \max_{1 \leq i \leq N} (2\gamma_1 c + \gamma_1 r_i^\alpha)$ is $r_i = \frac{L}{N}$, for $i = 1, 2, \dots, N$, 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 $\mathbb{E}_{single}(L, N, \gamma) = \gamma[2c + (\frac{L}{N})^\alpha]$.

Proof: A detailed proof can be found in Appendix B. \square

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 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

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

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 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. \square

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 $V = \{v_0, v_1, \dots, v_M, v_{M+1}, \dots\}$, $E = \{e_1, e_2, \dots\}$. Let $v_i = s_i$ for $i = 0, 1, \dots, M$. Here, vertices v_j , $j \geq M + 1$, are called *merge vertices* whose function will be explained later. Let e_1, e_2, \dots 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 λ_{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

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

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

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

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 determined². The R-node number on each edge can be computed 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 $\sqrt[\alpha]{\lambda_{e_i}}$.

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

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
 cost $\mathbb{E}_{e_i} = \mathbb{E}_{single}(L_{e_i}, n_{e_i}, \lambda_{e_i})$;
 $n_{e_i} \leftarrow n_{e_i} + 1$; $N \leftarrow N - 1$;
4: **end while**;
5: **return** n_{e_i} for all $e_i \in E$;

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

The above problem is a generalized version of the Euclidian Steiner Minimum Tree problem³, which is NP-hard [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 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 1 to $M - 1$. Appendix C shows the details of the hybrid algorithm. 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.

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 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.

3. 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.

5.2.1 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), 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 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. \square

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.

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.

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. We adopt similar configurations from [9][27][17] 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 other 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 5.2. It is the most straightforward approach and serves as a base-line. *Connectivity-Only* is chosen from a series of state-of-the-art schemes proposed in [15][27], which optimize the system performance by considering connectivity only. 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*

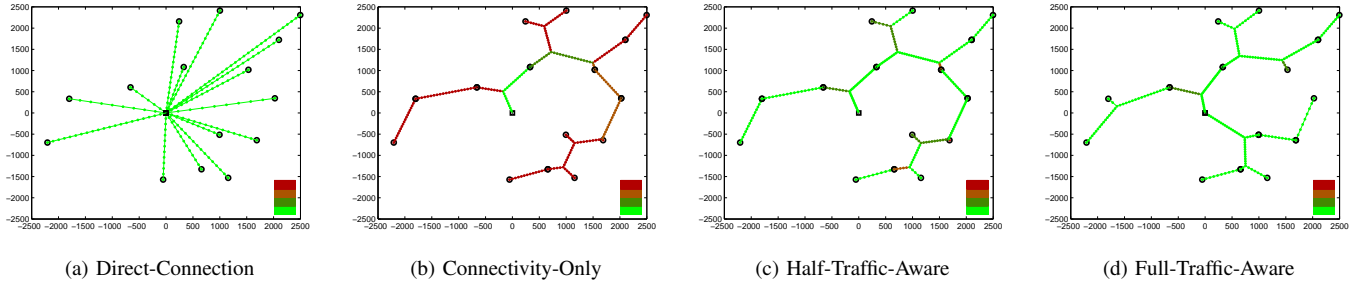


Fig. 6: An illustration of different deployment approaches and their residual energy distributions at the end of the network lifetime of each approach. The sink is denoted by the small square at the center. S-nodes are denoted by small circles. R-nodes are denoted by small diamond dots. Each approach 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.

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 $R_{max} = 500m$. 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. The formulas used for our numerical analysis can be found in Appendix E. In the following subsections, 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 $5000m \times 5000m$ with $R_{max} = 500m$, 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 4 R-nodes and maximally 8 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 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 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 seem 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 5 times of Connectivity-Only, Full-Traffic-Aware further raises the gain up to 15 times and 7 times, respectively, which is 40% 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 R-node assignment and merge vertex adjustments).

Fig. 9 shows the results of the total residual energy with different number of R-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. This also explains that the residual energy of Connectivity-Only increases much faster than the other

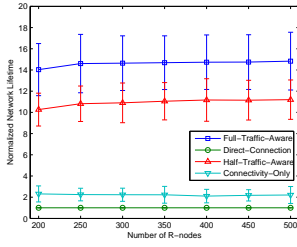


Fig. 7: Normalized network lifetime with different number of R-nodes by numerical analysis.

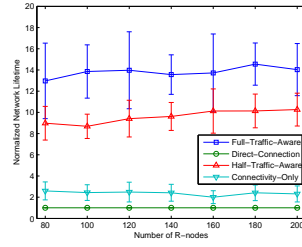


Fig. 8: Normalized network lifetime with the number of R-nodes ≤ 200 by numerical analysis.

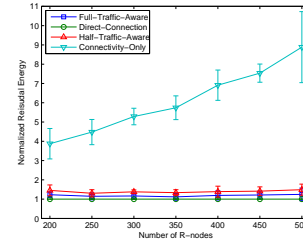


Fig. 9: Normalized residual energy with different number of R-nodes by numerical analysis.

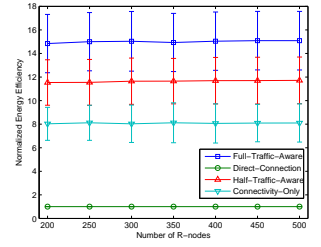


Fig. 10: Normalized energy efficiency with different number of R-nodes by numerical analysis.

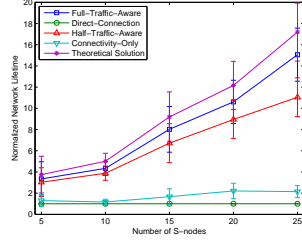


Fig. 11: Normalized network lifetime with different number of S-nodes by numerical analysis.

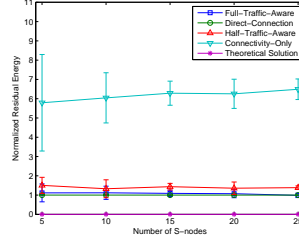


Fig. 12: Normalized residual energy with different number of S-nodes by numerical analysis.

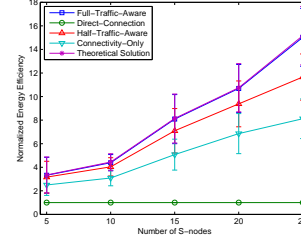


Fig. 13: Normalized energy efficiency with different number of S-nodes by numerical analysis.

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 amount 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 R-node 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). 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 Eq. 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 ($M \leq 15$) that can be verified within acceptable timeframes, 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%, and it 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% 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. 6a, Fig. 6c and Fig. 6d. In addition, there are still several edges with the residual energy more than 50% 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 fourth 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 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 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 only solutions cannot be directly applied here. We then 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 that for all test cases where the continuous space optimal solution can be computed within acceptable timeframes, 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 3-D 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.

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] E. L. Lloyd and G. Xue, "Relay Node Placement in Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 56, no. 1, pp. 134–138, January 2007.
- [16] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," in *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," in *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," in *IEEE INFOCOM*, 2006.
- [19] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transit Power Adjustment," in *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," in *ACM SenSys*, 2007.
- [21] S. Singh, M. Woo, and C. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," in *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," 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-tolerant Relay Node Placement in Wireless Sensor Networks: Problems and Algorithms," in *IEEE INFOCOM*, 2007.



Feng Wang (S'07) received both his Bachelor's degree and Master's degree in Computer Science and Technology from Tsinghua University, Beijing, China, in 2002 and 2005, respectively. He is currently a Ph.D. student in School of Computing Science at 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 IEEE and IEEE Communications Society since 2007. In 2006 summer, he interned in Wireless and Networking Group at Microsoft Research Asia, where he conducted research on peer-to-peer live video streaming. He also conducted research in 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 Chinese Academy of Sciences, where he worked on the testbed and information system design for wireless sensor networks. He was awarded Tsinghua University Scholarship for Excellent Student in 1998, 2000 and 2001. At 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 Self-financed Students Studying Abroad in 2009.



Dan Wang (S'05-M'07) received the B. Sc degree from Peking University, Beijing, China, in 2000, the M. Sc degree from Case Western Reserve University, Cleveland, Ohio, USA, in 2004, and the Ph. D. degree from Simon Fraser University, Burnaby, B.C., Canada, in 2007; all in computer science. He is currently an assistant professor at 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 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 co-recipient of the Best Student Paper Award of IWQoS'2008, the Best Paper Award (2009) of IEEE ComSoc Multimedia Com-

munications 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 IEEE and a member of Sigma Xi. He is an Associate Editor of IEEE Transactions on Multimedia, an Editor of IEEE Communications Surveys and Tutorials, and an Area Editor of Computer Communications. He is TPC Vice Chair for Information Systems of IEEE INFOCOM'2011.

APPENDIX A

LIST OF NOTATIONS

| Notation | Description |
|--------------------------|---|
| M | Number of given S-nodes. |
| N | Number of given R-nodes. |
| R_{max} | Maximum communication range of R-nodes. |
| α | Exponent parameter in communication model. |
| c | Small constant specified by physical features of transceiver in communication model. |
| s_0 | Location of remote data sink. |
| s_i | Location of S-node i . |
| γ_i | Average data rate of S-node i . |
| f_i | Location of R-node i . |
| \overline{xy} | Distance between node x and y . |
| r_i | Communication range of R-node i . |
| $\mathbb{E}_{subscript}$ | Energy cost of an R-node under the situation described by the <i>subscript</i> . |
| V | Vertex set in constructed graph $G(V, E)$. |
| E | Edge set in constructed graph $G(V, E)$. |
| v_i | Vertex i in V . $v_i = s_i$ for $0 \leq i \leq M$; otherwise v_i is a merge vertex. |
| e_i | Edge i in E , which is directed and traffic flows can only pass it following its direction. |
| L_{e_i} | Length of e_i . |
| λ_{e_i} | Total data rate of traffic flows passing e_i . |
| n_{e_i} | Number of R-nodes assigned on e_i . |
| \mathbb{E}_{e_i} | Maximum energy cost of one R-node on e_i . |
| \mathbb{E} | Initial energy budget on one R-node. |
| T | Network lifetime. |

APPENDIX B

PROOFS OF THEOREMS

B.1 Proof of Theorem 1

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

Proof: It is easy to prove the theorem by contradiction. Assume in the optimal solution, $\exists r_i \neq \frac{L}{N}$, then there must exist one, say r_j , which is greater than $\frac{L}{N}$. Then for this R-node, its energy cost is greater than the one given by the theorem, which contradicts with the assumption at the beginning and thus proves the theorem. \square

B.2 Proof of Theorem 2

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: By the discussion in Section 4.2, the situations for two traffic flows that either merge or separate completely have been proved.

For other situations of two traffic flows (i.e., two flows partially merge/separate), the theorem can be proved by contradiction. Assume that the optimal solution better than merging two flows completely exists. Then it is easy to divide

the whole flow paths into segments such that within any segment, the two flows either completely merge or separate. By the proof for the situations of completely merging or separating, in each segment where the two flows separate completely, the two flows can be merged together without sacrificing the network lifetime until the two flows in all segments are merged completely. This contradicts with that the assumed optimal solution is better than merging two flows completely and completes the proof for two traffic flows.

For the case of K traffic flows, the proof can be done by complete induction. The basic case is two traffic flows, which has been proved above. For the advance case of K flows, we assume that the theorem holds for all cases from 2 to $K - 1$ flows and show it also holds for the case of K flows. The proof is similar to the proof of the two-flow case. First, it is easy to prove that merging all K traffic flows completely is better than leaving them completely separated. For the remaining situations, we use contradiction and assume the optimal solution better than merging K flows completely exists. Then it is easy to divide the whole flow paths into segments such that within any segment, each pair of flows are either completely merged or separated. Now in every section, we consider each pair of completely merged flows as one new flow until all remaining flows are completely separated. Clearly the resulting flows in each section can be merged into one without sacrificing the network lifetime (either due to induction hypothesis or previously proved that merging all K traffic flows completely is better than leaving them completely separated) until all sections have only one flow left. This contradicts with that the assumed optimal solution is better than merging K flows completely and finishes the proof for the theorem. \square

It is worth noting that Theorem 2 and its proof also indicate that in the optimal solution, there is no need to split one single flow into multiple flows, since as shown in the proof of Theorem 2, merging the (split) multiple flows back into one flow achieves the optimal result.

B.3 Proof of Theorem 3

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: We prove it by induction on the number of given R-nodes N . For basic case, we have $N = |E|$. The optimal and only feasible assignment is $n_{e_i} = 1$ for all $e_i \in E$.

Now we assume that the assignment achieved by our algorithm is optimal for case $N = k$ with $k \geq |E|$ and consider case $N = k + 1$. We show the optimality by contradiction.

Assume that there is a better assignment which has lower maximum energy cost among the edges than our assignment. Let e_i be the edge that has the maximum energy cost in our assignment, and e_j be the edge that has the maximum energy cost in the new optimal assignment. We add a “/” on the notations used for the new optimal assignment. Then we have $\mathbb{E}_{e_i} > \mathbb{E}'_{e_j} \geq \mathbb{E}'_{e_i}$, which follows $n_{e_i} < n'_{e_i}$. This means that

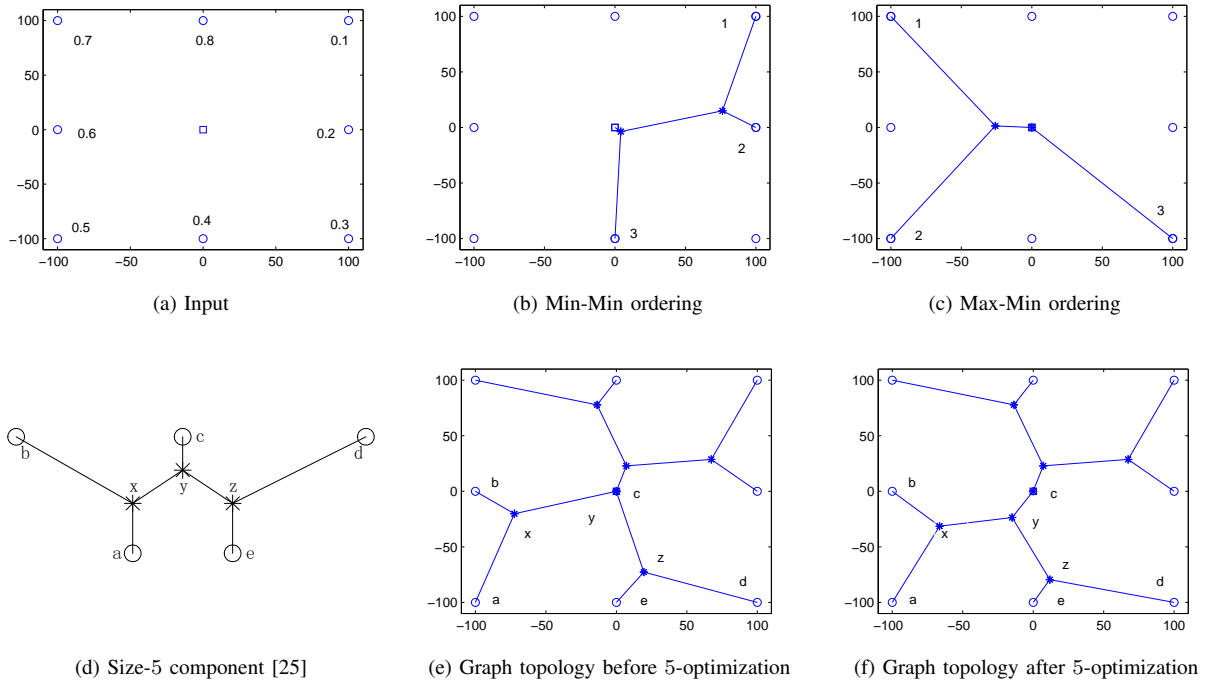


Fig. 14: An illustration of different orderings and 5-optimization used in the hybrid algorithm. The sink is denoted by the small square at the center. S-nodes are denoted by small circles. Merge vertices are denoted by small stars. (a) shows the input of the illustration, where the numerical label beside each S-node shows the data rate. (b) shows the first three steps by using Min-Min ordering, where the digit beside an S-node shows in which step the node is added. (c) shows the first three steps by using Max-Min ordering, where the digit beside an S-node shows in which step the node is added. (d) shows the pattern of the size-5 component used for optimization. (e) shows an intermediate graph topology before a 5-optimization and (f) shows the result after the 5-optimization, where the positions of x , y and z have been optimized within the size-5 component.

to make our assignment optimal, at least 1 R-node need to be moved from some other edge, say e_x to e_i while still keeping $\mathbb{E}_{e_x}^- \leq \mathbb{E}_{e_j}' \leq \mathbb{E}_{e_i}$. Note we use “ $\mathbb{E}^-/\mathbb{E}^+$ ” to denote the energy cost after an R-node is removed/added.

Now we consider by our R-node assignment algorithm, which edge has been assigned the $(k+1)$ -th R-node from case $N = k$ to case $N = k+1$. We argue it must be e_i under the above situation. Otherwise, if it is some edge e_y other than e_i , then instead of adding the $(k+1)$ -th R-node to e_y , we can move the extra R-node from e_x to e_y . By doing this, e_i now have the maximum energy cost other than e_y for case $N = k$ (note $\mathbb{E}_{e_i} > \mathbb{E}_{e_x}^-$ and $\mathbb{E}_{e_i} \geq \mathbb{E}_{e_y}^+$ due to case $N = k+1$), which means case $N = k$ can be further improved and contradicts with that case $N = k$ is optimal. Thus the $(k+1)$ -th R-node must be assigned to e_i . However, if now we move the extra R-node from e_x to e_i instead of assigning the $(k+1)$ -th R-node, we get a better assignment for case $N = k$ which also contradicts with that case $N = k$ is optimal. This shows the assumption that an assignment better than ours for case $N = k+1$ exists is not correct. And together with the basic case, the theorem is proved. \square

APPENDIX C HYBRID ALGORITHM FOR GRAPH TOPOLOGY MINIMIZING TOTAL WEIGHTED EDGE LENGTH

Our hybrid algorithm for computing the graph topology that minimizes the total weighted edge length is based on con-

ducting 5-optimization with size-5 component. Fig. 14 shows an example of the size-5 component and how to use it for 5-optimization. The size-5 component is a steiner-tree-like structure containing 5 outer vertices (a , b , c , d and e as illustrated in Fig. 14) and 3 inner vertices (x , y and z as illustrated in Fig. 14). During a 5-optimization, the pattern of the size-5 component is matched iteratively on a graph topology and the 3 inner vertices are optimized within the component as they are merge vertices while 4 of the 5 outer vertices are deemed as non-merge vertices with the remaining one serving as the sink. Fig. 15 shows our hybrid algorithm that uses both Min-Min and Max-Min orderings (see Fig. 14b and Fig. 14c) complementarily to bypass local optimums and minimizes the total weighted edge length.

APPENDIX D MERGE VERTEX ADJUSTMENT ALGORITHMS

We also 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.

EnergyBalance() (Fig. 16) proceeds iteratively (the **while** loop) to balance energy consumption among edges connecting to each merge vertex. In each iteration (the **for** loop), it tries

Algorithm MinWeightedLength()

```

1:  $L_{min} \leftarrow \infty$ ;
2:  $V_+ \leftarrow \{v_0\}$ ;  $E_+ \leftarrow \{\}$ ;  $V_- \leftarrow \{v_0\}$ ;  $E_- \leftarrow \{\}$ ;
3: for  $k = 1 \dots M - 1$ , do
4:   for  $i = 1 \dots M$ , do
5:     if  $i \leq k$ ,
6:       update  $V_+, E_+$  by adding one non-merge
       vertex that minimizes the increased total
       weighted length;
7:       update  $V_-, E_-$  by adding one non-merge
       vertex that maximizes the minimum
       increased total weighted length;
8:     else
9:       update  $V_+, E_+$  by adding one non-merge
       vertex that maximizes the minimum
       increased total weighted length;
10:      update  $V_-, E_-$  by adding one non-merge
       vertex that minimizes the increased total
       weighted length;
11:    end if
12:  end for
13:  conduct 5-optimization on  $(V_+, E_+)$  with each
  size-5 component;
14:  if the total weighted length  $< L_{min}$ ,
15:     $L_{min} \leftarrow$  the total weighted length;
16:     $V_{min} \leftarrow V_+$ ;  $E_{min} \leftarrow E_+$ ;
17:  end if
18:  conduct 5-optimization on  $(V_-, E_-)$  with each
  size-5 component;
19:  if the total weighted length  $< L_{min}$ ,
20:     $L_{min} \leftarrow$  the total weighted length;
21:     $V_{min} \leftarrow V_-$ ;  $E_{min} \leftarrow E_-$ ;
22:  end if
23: end for
24: return  $V_{min}$  and  $E_{min}$ ;

```

Fig. 15: The hybrid algorithm for computing the graph topology that minimizes the total weighted edge length.

to adjust the location of a merge vertex v by solving equations

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

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 experience shows that the algorithm is fast in practice.

AdjustMergeVertex() (Fig. 17) takes the graph topology generated by the theoretical solution as an input. It first assigns R-nodes and does energy balancing (line 1-3). 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 9-10), so as to bypass local optimums.

APPENDIX E FORMULAS FOR NUMERICAL ANALYSIS

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

Algorithm EnergyBalance()

```

1:  $V_{min} \leftarrow V$ ;  $\mathbb{E}_{min} \leftarrow \max_{e_i \in E} \mathbb{E}_{e_i}$ ;
2: while true, do
3:   for each merge vertex  $v \in V$ , do
4:     adjust  $v$  so as to balance  $\mathbb{E}_{e_i}$  among all  $e_i$ 
     connecting to  $v$ ;
5:   end for
6:   if  $\max_{e_i \in E} \mathbb{E}_{e_i} < \mathbb{E}_{min}$ ,
7:      $V_{min} \leftarrow V$ ;  $\mathbb{E}_{min} \leftarrow \max_{e_i \in E} \mathbb{E}_{e_i}$ ;
8:   else break;
9: end while;
10: return  $V_{min}$ ;

```

Fig. 16: The algorithm for balancing energy consumption among different edges.

Algorithm AdjustMergeVertex()

```

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

```

Fig. 17: The algorithm for merge vertex adjustment.

estimated as⁴

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

The total residual energy is

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

And the energy efficiency is

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

With the given parameter setting (the initial energy budget \mathbb{E} , required minimum network lifetime T_{min} , maximum communication range R_{max} , the number of S-nodes M and their locations s_i and data rates γ_i), the upper bound of the minimum R-node number N for Direct-Connection can be

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

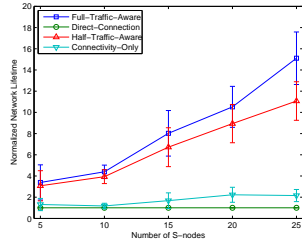


Fig. 18: Normalized network lifetime with different number of S-nodes by $ns-2$ simulations.

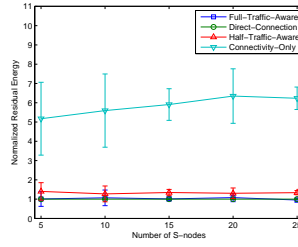


Fig. 19: Normalized residual energy with different number of S-nodes by $ns-2$ simulations.

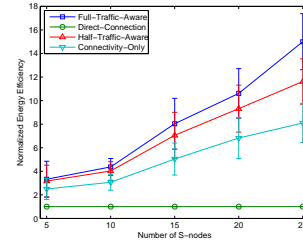


Fig. 20: Normalized energy efficiency with different number of S-nodes by $ns-2$ simulations.

derived by

$$\left\{ \begin{array}{l} \left(\sum_{i=1}^M \sqrt{\gamma_i} \cdot \overline{s_i s_0} \right) / \overline{N} \leq \sqrt{\frac{\mathbb{E}}{T_{min}}} \\ n_i = \left\lfloor \frac{\overline{N} \cdot \sqrt{\gamma_i} \cdot \overline{s_i s_0}}{\sum_{j=1}^M \sqrt{\gamma_j} \cdot \overline{s_j s_0}} \right\rfloor \\ \max_{i=1 \dots M} \left(\frac{\overline{s_i s_0}}{n_i} \right) \leq R_{max} \\ N = \sum_{i=1}^M n_i \end{array} \right.$$

APPENDIX F SIMULATION RESULTS ON $ns-2$

To further evaluate our solution, we conducted extensive simulations on $ns-2$, 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 [18] as in previous sections, i.e., $\alpha = 4$ and $c = 4500$ for both sending and receiving.

Fig. 18, Fig. 19 and Fig. 20 show the results of network lifetime, residual energy and energy efficiency with different

number of S-nodes, respectively. It is easy to see that our Full-Traffic-Aware solution performs the best and achieves up to 7 times of the lifetime of Connectivity-Only and 15 times of Direct-Connection when the number of S-nodes increases to 25. On the other hand, the residual energy of our Full-Traffic-Aware solution stays almost as low as Direct-Connection and is significantly less than the other two schemes, i.e., Connectivity-Only and Half-Traffic-Aware schemes. Moreover, Full-Traffic-Aware also has the best energy efficiency and when the number of S-nodes achieves 25, the average amount of traffic delivered by per unit energy cost through our Full-Traffic-Aware solution is nearly 90% more than the Connectivity-Only scheme and is about 15 times of the amount through the Direct-Connection scheme.

Comparing with the results by numerical analysis, it is easy to find that the simulation results match the 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 numerical 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. 6a, 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.