A Shortest Path Tree Based Algorithm for Relay Placement in a Wireless Sensor Network and its Performance Analysis☆

Abhijit Bhattacharya[a], Anurag Kumar[a]

[a]*Dept. of Electrical Communication Engineering*
Indian Institute of Science, Bangalore, 560012, India

Abstract

In this paper, we study a problem of designing a multi-hop wireless network for interconnecting sensors (hereafter called *source* nodes) to a Base Station (BS), by deploying a minimum number of relay nodes at a subset of *given potential locations*, while meeting a quality of service (QoS) objective specified as a hop count bound for paths from the sources to the BS. The hop count bound suffices to ensure a certain probability of the data being delivered to the BS within a given maximum delay under a light traffic model. We observe that the problem is NP-Hard. For this problem, we propose a polynomial time approximation algorithm based on iteratively constructing shortest path trees and heuristically pruning away the relay nodes used until the hop count bound is violated. Results show that the algorithm performs efficiently in various randomly generated network scenarios; in over 90% of the tested scenarios, it gave solutions that were either optimal or were worse than optimal by just one relay. We then use random graph techniques to obtain, under a certain stochastic setting, an upper bound on the average case approximation ratio of a class of algorithms (including the proposed algorithm) for this problem as a function of the number of source nodes, and the hop count bound. To the best of our knowledge, the average case analysis is the first of its kind in the relay placement literature. Since the design is based on a light traffic model, we also provide simulation results (using models for the IEEE 802.15.4 physical layer and medium access control) to assess the traffic levels up to which the QoS objectives continue to be met.

Keywords: wireless sensor networks, QoS based design of wireless sensor networks, relay placement for wireless sensor networks, design of multi-hop CSMA networks, node-weighted Steiner tree, hop constrained Steiner tree

1. Introduction

1.1. Motivation and Problem Definition

Large industrial establishments such as refineries, power plants and electric power distribution stations typically have a large number of sensors distributed over distances of hundreds of meters from the control center. Individual wires carry the sensor readings to the control center. Recently there has been increasing interest in replacing these wireline networks with wireless packet networks ([1, 2, 3]). A similar problem arises in an intrusion detection application using a fence of passive infrared (PIR) sensors [4], where the event sensed by several sensors has to be conveyed to a Base Station (BS) quickly and reliably.

The communication range of the sensing nodes is typically a few tens of meters (depending on the RF propagation characteristics of the deployment region). Therefore, usually multi-hop communication is needed to transmit the sensed data to the BS. The practical problem that our work is aimed towards is the following:

1. There are already deployed, static sensors from which measurements, encapsulated into packets, need to be delivered to a single BS. We also refer to the sensors as *sources.*
2. Additional relays need to be placed in the region in order to provide multi-hop paths from the sources to the BS. The *sources can also act as relays* for the packets from other sources.
   In most practical applications, due to the presence of obstacles to radio propagation, or due to taboo regions, we cannot place relay nodes anywhere in the region, but only at certain designated locations. This leads to the

☆A preliminary version of this work, entitled *Delay Constrained Optimal Relay Placement for Planned Wireless Sensor Networks*, was presented in IEEE IWQoS 2010, Beijing, 2010. This work was supported by the Department of Electronics and Information Technology under the Automation Systems Technology (ASTEC) program, and by the Department of Science and Technology via a J.C. Bose Fellowship.

Email addresses: abhijit@ece.iisc.ernet.in (Abhijit Bhattacharya), anurag@ece.iisc.ernet.in (Anurag Kumar)

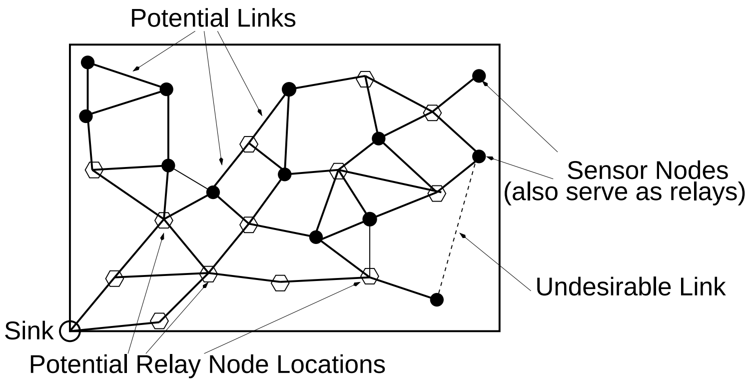

Figure 1: The constrained relay placement problem; circles indicate sources, and the hexagons indicate *potential* relay locations. The edges denote the useful links between the nodes.

problem of *constrained relay placement* in which the relays are constrained to be placed at certain *potential relay locations*. Furthermore, only certain links are permitted[1]. See Figure 1 for a depiction of the problem.

3. The objective of the design is to *place as few additional relays as possible* (at the potential relay locations) while achieving a network that meets the following requirements:
   (a) There is a path from each source node to the BS; i.e., we seek a tree that spans the source nodes, and is rooted at the sink[2].
   (b) The hop count from each source to the sink is at most $h_{\max}$. Under the assumption of light traffic and CSMA/CA MAC, this hop count bound ensures a stochastic QoS objective, namely that the maximum delay on any path is bounded by a given value $d_{\max}$, and the packet delivery probability (the probability of delivering a packet within the delay bound) on any path is at least p_{del}. See Section 2.2 and [5] for details.

1.2. An Overview of Our Approach and Contributions

We are concerned with the design of a QoS aware multi-hop CSMA/CA network for connecting wireless sensors with a sink, by selecting a small number of potential locations at which to place relays. In this paper we limit ourselves to the light traffic setting, which is adequate for modelling low arrival rates (say, one packet every few seconds from each source) that are typical of the so called *condition monitoring/industrial telemetry* applications. In this setting, analysis of the wireless physical layer and the medium access control used by the system (e.g., IEEE 802.15.4, which is used by Zigbee networks [6]) yields the following (see Sections 2.1 and 2.2 for a summary of the arguments):
(i) A notion of a "feasible" edge between a pair of nodes (sources or potential relays), thus yielding a graph over the sources and potential relay locations.
(ii) A hop count bound between each source and the sink, which ensures the stochastic QoS objective for packet delivery.

Thus, we arrive at a graph design problem with the requirements mentioned earlier in Section 1.1. After an extensive literature survey, we concluded that this problem of hop constrained, cost optimal network design has not yet been well studied. Although Sitanayah et al. [7, 8] have proposed heuristics for this problem for general k, they have not made any attempt at a formal study of the complexity of the problem, or the performance guarantees of their algorithms. Very recently, Nigam and Agarwal [9] proposed a branch-and-cut algorithm to solve only a *subclass* of this problem optimally. However, even for that subclass of problems, their algorithm is not polynomial time, and hence cannot be used for large sized problem instances. Voss [10] studied a related problem of hop constrained, minimum total edge-cost network design, and proposed tabu search based heuristics for the same; but again, no formal theoretical study of either the problem, or the proposed algorithm was presented. As there is a considerable amount of relevant literature to be discussed, we have placed our detailed survey of related literature in Section 7 just before the Conclusion section.

The overall approach we take to solving the problem is the following:

[1]This could be because some links could be too long, leading to a high bit error rate and hence large packet delay, or due to an obstacle, e.g., a firewall.

[2]We have also developed algorithms for the problem where the requirement is to have at least $k > 1$ paths from each source to the BS. See [5] for details.

1. Having converted the problem into a graph design problem, we analyze the complexity of this problem to show that the problem is NP-Hard, and develop approximation algorithms for this problem.
   The class of algorithms that we develop basically perform a series of shortest path computations from each source to the sink, starting with an initial feasible solution and adopting a certain combinatorial relay pruning strategy to prune relay nodes from the feasible solution sequentially; each time a relay node is pruned, a new shortest path is computed involving only the remaining nodes, while still retaining hop count feasibility. These algorithms are *simple, intuitive, and fast*, and we have found that they work very well (often yielding optimal or close to optimal solutions) in our extensive numerical exploration. For brevity, we describe and analyze only one such algorithm in this paper. For other algorithms of this class, see [11].
2. We have provided a worst case analysis of the proposed algorithm. Also, a bound on the average case approximation ratio of our algorithm has been derived for a particular stochastic setting using a random graph model. While the bound we obtain is loose when compared to numerical simulations, our analysis technique would be of independent interest, since *there does not seem to exist similar average case analysis for the relay placement/graph design algorithms proposed in the literature* (see Section 7).
3. Finally, in order to study the limits of the performance of our designs with "positive" traffic (i.e., for non-zero traffic arrival rates such that there may be contention in the network), we also provide packet level simulation results (using Qualnet, and assuming IEEE 802.15.4 CSMA/CA Medium Access Control) for the designs obtained using our algorithm (see Section 6).

1.3. Organization of the Paper

The rest of the paper is organized as follows: in Section 2 we formally describe the problem formulation, and discuss the complexity of the problem. In Section 3 we propose a polynomial time algorithm (SPTiRP) for the problem, and provide a worst case analysis of the algorithm. Section 4 provides an average case analysis of the SPTiRP algorithm under a certain stochastic setting. In Section 5 we provide extensive numerical results for the SPTiRP algorithm applied to a set of random scenarios. Section 6 provides packet level simulation results for the designs obtained using our algorithm. Finally, we present a detailed survey of closely related literature in Section 7, and conclude the paper in Section 8.

2. The Network Design Problem

In Sections 2.1 and 2.2 below, we discuss how we have converted the problem of designing a multi-hop CSMA/CA network with a stochastic QoS objective into a graph design problem. The resulting graph design problem is formally defined in Section 2.3, and the remaining subsection discusses the complexity of the problem.

2.1. The Lone Packet Model

In this paper, we address the problem for the case where the traffic from the source nodes is *light*. Formally, we are concerned with the situations in which the traffic is so light that at any point of time, only one measurement packet travels from a source in the network to the BS. From now on, we shall refer to this as the "lone packet traffic model," which is realistic for many applications where the time between successive measurements being taken is sufficiently long so that the measurements can be staggered so as not to occupy the medium at the same time. Such slow measurement rates are typical of so-called *condition monitoring/industrial telemetry* applications [12, 13].

Moreover, it can be shown that for a CSMA/CA network to satisfy a probability of delivery objective for a given positive traffic arrival rate, it is *necessary* that the network satisfies this QoS objective under the lone packet model (for a formal proof[3] of this fact, see [5]). As we shall see in subsequent sections, even under this lone packet model, the problem of QoS constrained network design is computationally hard, and it does not seem to have been well studied (see Section 7). We cannot hope to solve the general problem of QoS aware network design for arbitrary positive traffic arrival rates unless we have a reasonably good solution to the more basic problem of lone packet based network design.

2.2. The Network Design Setting

In this subsection, we discuss how we can map the packet level QoS objectives into graph level objectives under the lone packet model. Given a set of source nodes or required vertices Q (including the BS) and a set of potential relay locations R (also called Steiner vertices), we consider a graph $G = (V, E)$ on $V = Q \cup R$ with E consisting of all *feasible* edges.

[3] A formal proof is necessary for this seemingly obvious statement since, in CSMA/CA networks, in general, the performance is not monotone with the arrival rates (see, e.g., [14]); hence, the statement about the lone-packet traffic model needs to be made with care.

Note that there are several ways in which we can define the set of *feasible* edges E, keeping in mind the end-to-end QoS objective. For example, we can impose a bound on the packet error rate (PER) of each link, or alternately, we can constrain the maximum allowed link length (which, in turn, affects the link PER). Having thus characterized the link quality of each feasible link in the graph G, it can be shown by an elementary analysis that the QoS objectives ($d_{\max}$ and p_{del}) can be met by imposing a hop count bound of $h_{\max}$ between each source node and the sink. Details of this analysis are provided in the technical report [5], where we have considered the practical situation of slowly fading links, and packet losses due to random channel errors. Thus, there is a random delay at each hop due to packet retransmissions, and packets could be dropped if a retransmission limit is reached. Note that as a consequence of the lone packet assumption, the delay along a path is additive, i.e., it is simply the sum of the delays on each hop along the path.

Note that in this paper we do not address the important practical issues of actually identifying the usable links on the field (i.e., the edge set E of the on-field network graph G), or handling their variations over time. Those details and other implementation issues are available in [16]. Furthermore, the graph design algorithms presented in this paper are in no way tied to the link modelling approach mentioned above for defining the graph G, and the hop constraint $h_{\max}$. The algorithms can be applied as long as a graph on $Q \cup R$, and a hop constraint is given, irrespective of how the graph and the hop constraint were obtained.

Finally, throughout this paper, we assume that the potential relay locations are given to us. In case the user specifies only certain available regions instead of the exact locations, there are several ways in which the potential locations can be obtained. One possible random selection approach which ensures a feasible solution with high probability has been prescribed in Section 4 in the context of the average case analysis of our algorithm.

2.3. Problem Formulation

Given the graph $G = (V, E)$ on $V = Q \cup R$ with E consisting of all *feasible* edges (as explained in Section 2.2), and a hop constraint $h_{\max}$, *the problem is to extract from this graph, a spanning tree on Q, rooted at the BS, using a minimum number of relays such that the hop count from each source to the BS is at most* $h_{\max}$. We call this the *Rooted Steiner Tree-Minimum Relays-Hop Constraint* (RST-MR-HC) problem.

2.4. Complexity of the Problem

Proposition 1. *1. The RST-MR-HC problem is NP-Hard.*
2. It cannot be approximated to a factor better than $O(\log m)$, where m is the number of sources.

Proof. 1. The subset of RST-MR-HC problems where the hop count bound is trivially satisfied is precisely the class of *node-weighted Steiner tree* (RST-MR) [17] problems (see our literature survey in Section 7 for a discussion of the node-weighted Steiner tree (RST-MR) problem). Consider, for example, all RST-MR-HC problems where $|Q| + |R| = N$, N being some positive integer, and the hop count bound is $h_{max} = N - 1$. Clearly, the hop count bound is trivially satisfied in these problems. Thus, the RST-MR problem is a subclass of the RST-MR-HC problem. But, the RST-MR problem is NP-Hard (see [17]). Hence, the RST-MR-HC problem is also NP-Hard by restriction argument[18, p. 63, Section 3.2.1].

2. It was proved in [17], by a reduction from the set cover problem, that the node-weighted Steiner tree or RST-MR problem cannot be approximated to a factor better than $O(\log m)$, where m is the number of sources. It follows that the RST-MR-HC problem, being a superclass of the RST-MR problem, cannot be approximated to a factor better than $O(\log m)$. □

In the literature, the *edge-weighted* Steiner tree problem frequently arises in the context of the design of *wireline* telecommunication networks, as in such networks the cost of the network includes the cost of laying the links. On the other hand, in a multihop wireless network, the cost of the network is just the cost of installing the relay nodes, which leads to the *node-weighted* Steiner tree problem. While certain variations of the edge-weighted Steiner tree problem are constant factor approximable, Proposition 1 shows that the node-weighted Steiner tree problem (*even without hop constraint*) cannot be approximated to a factor better than $O(\log m)$. The following proposition shows that the edge-weighted Steiner tree problem is, in fact, a subclass of the node-weighted Steiner tree problem.

Proposition 2. *The node-weighted Steiner tree problem with hop constraint (resp., diameter bound)* includes, as special cases*, the edge-weighted Steiner tree problem with hop constraint (resp., diameter bound), and the hop constrained (resp., bounded diameter) minimum spanning tree problem*[4].

[4]The reason for considering this variation of the problem will be clear when we discuss related literature in Section 7.

Proof. We present the proof for the hop constrained case. The proof for the diameter bounded case is essentially identical.

First notice that the hop constrained minimum spanning tree problem is a special case of the hop constrained edge-weighted Steiner tree problem (where the set of Steiner vertices is empty). Thus, it suffices to prove the result for the hop constrained edge-weighted Steiner tree problem. The problem is stated as: given a graph $G = (V, E)$ with non-negative weights $c(e)$ associated with each edge $e \in E$, a subset of vertices $Q \subset V$ along with a designated root node or sink, and a hop constraint $h_{\max}$, find a tree with minimum total edge weight, rooted at the sink, and spanning Q, such that the path from each vertex in Q to the sink has hop count at most $h_{\max}$.

Given any instance of the hop constrained edge-weighted Steiner tree problem, we can convert it into an instance of the hop constrained node-weighted Steiner tree problem as follows (this is a minor variation of a similar reduction presented in [19, p. 77, Section 4.1.1]): on each edge $e \in E$ of the graph G, we introduce a new Steiner vertex, and assign it a cost $c(e)$, the cost of the edge. Note that this splits the edge into two edges. Notice that the new Steiner vertices have degree exactly two, and a new Steiner vertex is used if and only if both the edges obtained from the original edge are used. In the resulting graph, all other Steiner vertices are assigned zero cost, and all edges are assigned zero cost. Note that this procedure does not affect the set Q. We now set the hop constraint to $2h_{\max}$. Then, it is easy to verify that on this modified graph, an optimal solution to the hop constrained node-weighted Steiner tree problem corresponds to an optimal solution to the hop constrained edge-weighted Steiner tree problem in the original graph G, and vice-versa.

Since the instances resulting from the above reduction are a strict subset of the class of all possible instances of the hop constrained node-weighted Steiner tree problem, the claim follows. □

3. RST-MR-HC: A Heuristic and its Worst Case Analysis

3.1. Shortest Path Tree (SPT) based Iterative Relay Pruning Algorithm (SPTiRP)

1. **The Zero Relay Case:** Let G_Q be the restriction of the graph G to the node set Q. Find an SPT on G_Q, rooted at the sink. If the hop count is at most $h_{\max}$ for each path, we are done; no relays are required in an optimal solution. Else, go to the next step.
2. Find a Shortest Path Tree T on G, rooted at the sink.
3. **Checking Feasibility:** If for any path in the SPT, the path cost exceeds $h_{\max}$, declare the problem infeasible. (Clearly, if the shortest path from a node to the sink does not meet the hop count bound, no other path from the node to the sink will meet the hop count bound). Else, go to the next step.

Pruning the SPT:

4. Discard all nodes in R that are not in T. Note that this step may lead to suboptimality as some of these discarded relay nodes could be part of an optimal solution.
5. For the remaining nodes in R, define the weight of a relay node as the number of paths in T that use that node.
6. Arrange the paths in T in increasing order of hop count.
7. Among the paths in T that use relay nodes, choose the least cost path, i.e., the one that has the least number of hops. This path has the maximum "slack" in the hop constraint. Arrange the relay nodes on this path in increasing order of their weights as defined in Step 5.
8. Remove the least weight relay node, and consider the restriction of G to the remaining nodes in T. Find an SPT on this graph. If in this SPT, path cost exceeds $h_{\max}$ for any path, then discard this SPT, replace the removed relay node, and repeat this step with the next least weight relay node. If all the relays in the least cost path have been tried without success, move on to the next least cost path, and repeat steps 7 and 8 for the relays in this path that have not yet been tried.
9. If in the above step, the SPT obtained satisfies the hop constraint for all the paths, then delete the removed relay node permanently from R, denote the newly obtained SPT by T, and repeat Steps 4 through 9.
10. Stop when no more relay pruning is possible without violating the hop constraint on one or more of the paths. Output T as the final solution.

Remarks:

1. Step 1 of the above algorithm ensures that if the optimal design does not use any relay node, then the same holds true for our algorithm. That way we can make sure that the algorithm does not do infinitely worse in the sense that $\frac{\text{Relay}_{\text{algo}}}{\text{Relay}_{\text{opt}}}$ is finite.

   The idea behind Steps 7, 8 and 9 is that choosing to remove a relay from the path with the most slack in cost (i.e., hop constraint), we stand a better chance of still meeting the hop count requirement with the remaining relays. Also, removing a relay of less weight would mean affecting the cost of a small number of paths. So by pruning relays in the manner as described in Steps 7, 8 and 9, we aim for a better exploration of the search space.

2. Note that the SPTiRP algorithm (and the worst case analysis presented next) can be applied to arbitrary input graphs in any dimensions, as opposed to geometric graphs in two dimensions considered in most of the previous work (see, for example, [20, 21, 22, 23]).
3. Recall that our objective is to select a minimal subset of the relays $R' \subseteq R$, and a topology on $Q \cup R'$, so that on the subgraph thus obtained there is a path from each source to the sink, each path meeting the QoS objective under the lone packet model. The SPTiRP algorithm in this section is a *network design algorithm* aimed at achieving this objective. Once we obtain a network topology using this design algorithm, any routing algorithm that can select one of the QoS-satisfying routes, from each source to the sink, statically or dynamically, can be used. For example the routing algorithm can dynamically select one of the possible QoS-satisfying routes from each source, so as to balance the relay energy consumption over time, or a standard routing protocol such as RPL (with an appropriately defined objective function) can be used.

Numerical experiments demonstrating the performance of the SPTiRP algorithm compared to the optimum solutions are presented in Section 5.2.

3.2. Analysis of SPTiRP

3.2.1. Complexity

The complexity of determining the shortest path tree on N nodes is $O(N \log N)$ [24]. Let us denote this function by $g_{\mathrm{SPT}}(\cdot)$. In Iteration 1 of the algorithm, the complexity is $g_{\mathrm{SPT}}(|Q|)$ and in Iteration 2, it is $g_{\mathrm{SPT}}(|Q| + |R|)$. In subsequent iterations, we remove one relay node at a time and find the SPT on the resultant restricted graph; if no improvement is found, we replace that node and continue. Thus, for the k^{th} iteration, the worst case complexity will be $(|R| - k + 3)g_{\mathrm{SPT}}(|Q| + |R| - k + 2)$, where in the worst case, $k = 3, 4, \ldots, |R| + 1$. Let $g_{\mathrm{sptirp}}(\cdot)$ denote the overall complexity. Thus, the overall complexity will be

$$\begin{aligned} g_{\mathrm{sptirp}}(|Q| + |R|) = g_{\mathrm{SPT}}(|Q| + |R|) + \\ & \sum_{j=1}^{|R|} (g_{\mathrm{SPT}}(|Q| + |R| - j))(|R| - j + 1) \\ & \leq (1 + |R|^2)(g_{\mathrm{SPT}}(|Q| + |R|)) \end{aligned}$$

which is polynomial time.

3.2.2. Worst Case Approximation Factor

Theorem 1. *The worst case approximation guarantee for the SPTiRP algorithm is* $\min\{m(h_{\max} - 1), (|R| - 1)\}$*, where m is the number of sources,* $h_{\max}$ *is the hop constraint, and* $|R|$ *is the number of potential relay locations.*

Proof. The worst case occurs when the SPT obtained before we enter Step (4) does not contain any relay node(s) that correspond to some optimal design. If no relays are used in any optimal design, then the algorithm will yield an optimal design (Step (1)). If an optimal solution uses a positive number of relays but not all of them, then SPTiRP cannot stop by using all the relays. Indeed, suppose that SPTiRP stops and uses all the relays. Since there is a feasible tree containing a strict subset of the relays, the pruning steps in SPTiRP will succeed in pruning at least one relay. Hence, the worst possibility is that the optimal design uses just 1 relay node, whereas the SPT obtained in Step (2) consists of all the remaining $(|R| - 1)$ relays, and moreover, pruning any of these $(|R| - 1)$ relays will cause one or more paths in the resulting SPT to violate the hop constraint. Thus, in the worst case, the algorithm may lead to a design with $(|R| - 1)$ relays instead of the optimal design with one relay. Also note that for a problem with m sources, and a hop constraint $h_{\max}$, no feasible solution can use more than $m(h_{\max} - 1)$ relays. Hence, we have a polynomial factor worst case approximation guarantee of $\min\{m(h_{\max} - 1), (|R| - 1)\}$. □

3.2.3. Sharp Examples (for Worst Case Approximation and for Optimality)

Let us now present *a sequence of problems of increasing complexity for which the approximation guarantee is strict*, i.e., for these problems, the algorithm ends up using $|R| - 1$ relays, while the optimum design uses one relay. Such examples are worthwhile to explore as they help to show that the approximation factor obtained above cannot be improved. Consider the situation shown in Figure 2. The green hexagons denote the relay node locations and the black circles represent the source node locations. Only the edges shown (coloured or black) are permitted. Consider the RST-MR-HC problem on this graph with $h_{max} = 3$. Clearly the optimal solution will use only one relay, R1, to reach from each source to the BS within the specified hop count bound. The black dotted links correspond to the optimal solution. The red link between source S1 and the BS will belong to both the optimal solution and the outcome of our algorithm as it is a direct link. Our SPT based algorithm will calculate the shortest paths and thus end up using relays R2, R3, . . . , Rn, leaving out R1. The black solid links correspond to the solution given by our algorithm. Clearly, in such problems, we end up using $|R| - 1$ relays instead of just one.

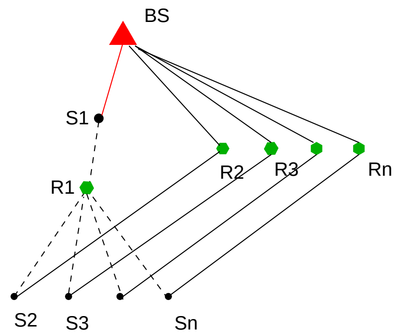

Figure 2: A sequence of problems where the worst case approximation guarantee is strict.

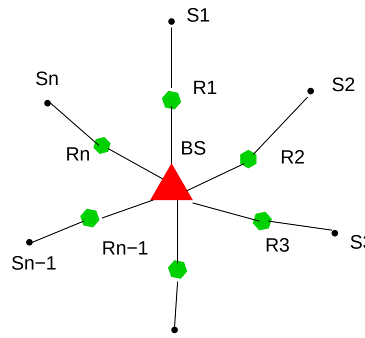

Figure 3: A sequence of problems where SPTiRP gives optimal solution.

Another sequence of problems of increasing complexity for which the algorithm gives the optimal design can be constructed as shown in Figure 3. Such examples help to show that the proposed algorithm does provide an optimal solution in some scenarios.

As before, the green hexagons represent relay locations and the black dots represent source nodes. Suppose $h_{max} = 2$. Then clearly, the optimal solution is as shown in the figure. The algorithm, after calculating the SPT, will end up with the same solution.

4. Average Case Analysis of SPTiRP

We shall derive below an upper bound on the average case approximation factor of SPTiRP in a certain stochastic setting. The derivation, in fact, applies to any algorithm that starts with an SPT, and proceeds by pruning relays from the SPT in some manner. The setting is chosen so as to ensure the existence of a feasible solution with high probability. For the purposes of this analysis, we restrict ourselves to two dimensional geometric graphs.

We consider a square area $A(\subset \Re^2_+)$ of side a. The BS is located at (0,0). We deploy n potential locations randomly over A, yielding the potential locations vector $\underline{x} \in A^n$. Then we place m sources over A, yielding source location vector $\underline{y} \in A^m$. Let $\omega = (\underline{x}, \underline{y})$, i.e., ω denotes the joint potential locations vector and source locations vector. We assume a model where a link of length at most r has the desired PER, and $h_{\max}$ is the hop constraint. We then consider the *geometric graph*, $\mathcal{G}^r(\omega)$, over these $n + m$ points; i.e., in $\mathcal{G}^r(\omega)$ there is an undirected edge between a pair of nodes in ω if the Euclidean distance between these nodes is at most r. If in this graph the shortest path from each source to the BS (at $(0, 0)$) has a hop count at most $h_{\max}$, then ω is *feasible*. Define

$H_j(\omega)$: Hop distance (i.e., the number of hops in the shortest path) of source j from the BS in $\mathcal{G}^r(\omega)$, $1 \le j \le m$. (∞ if source j is disconnected from the BS in $\mathcal{G}^r(\omega)$)

$\mathcal{X} = \{(\underline{x}, \underline{y}) : \forall y_j,\ 1 \le j \le m,\ H_j \le h_{\max}\}$: Set of all feasible instances

We would like $\mathcal{X}$ to be a high probability event. For this we need to limit the locations of the sources to be no more than $(1 - \epsilon)rh_{\max}$ from the BS; Theorem 2, later, will help characterize the relationship between ϵ, the number of potential locations, and the probability of $\mathcal{X}$.

For a given $\epsilon \in (0, 1)$, let $A_\epsilon (\subset A)$ denote the quarter circle of radius $(1-\epsilon)h_{\max} r$ centered at the BS, where $h_{\max}$ is the hop constraint, and r is the maximum allowed communication range.

Formally, we deploy n potential locations independently and identically distributed (i.i.d) uniformly randomly over the area A; then deploy m sources i.i.d uniformly randomly over the area A_ϵ. The probability space of this random experiment is denoted by $(\Omega^{(n)}_{m,\epsilon}, \mathcal{B}^{(n)}_{m,\epsilon}, P^{(n)}_{m,\epsilon})$, where,

$\Omega^{(n)}_{m,\epsilon} = (A^n \times A^m_\epsilon)(\subset \Re_+^{2(n+m)})$: Sample space; the set of all possible deployments

$\mathcal{B}^{(n)}_{m,\epsilon}$: The Borel σ-algebra in $\Omega^{(n)}_{m,\epsilon}$

$P^{(n)}_{m,\epsilon}$: Probability measure induced on $\mathcal{B}^{(n)}_{m,\epsilon}$ by the uniform i.i.d deployment of nodes

Consider the random geometric graph $\mathcal{G}^r(\omega)$ induced by considering all links of length at most r on an instance $\omega \in \Omega^{(n)}_{m,\epsilon}$. We introduce the following notation:

$N_{\text{SPTiRP}}(\omega)$: number of relays in the outcome of the SPTiRP algorithm on $\mathcal{G}^r(\omega)$ (∞ if $\omega \in \mathcal{X}^c$)

$R_{\text{Opt}}(\omega)$: number of relays in an optimal solution to the RST-MR-HC problem on $\mathcal{G}^r(\omega)$ (∞ if $\omega \in \mathcal{X}^c$)

The *average case approximation ratio* of the SPTiRP algorithm over feasible instances is defined as

$$\text{Average case approximation ratio, } \alpha \triangleq \frac{E[N_{\text{SPTiRP}}|\mathcal{X}]}{E[R_{\text{Opt}}|\mathcal{X}]} \tag{1}$$

Remark: This would be a useful quantity if the user of the algorithm wishes to apply the algorithm to several instances of the problem, yielding the required number of relays $N_1, N_2, \ldots, N_k$, as against the optimal number of relays $R_1, R_2, \ldots, R_k$, and is interested in the ratio $\frac{N_1+N_2+\cdots+N_k}{R_1+R_2+\cdots+R_k}$.

In the derivation to follow, we will need $\mathcal{X}$ to be a high probability event, i.e., with probability greater than $1-\delta$ for a given $\delta > 0$, the locations of the sources and potential relays form a feasible instance. The following result ensures that this holds for the construction provided earlier, provided the number of potential locations is large enough.

Theorem 2. *For any given $\epsilon, \delta \in (0, 1)$, $h_{\max} > 0$ and $r > 0$, there exists $n_0(\epsilon, \delta, h_{\max}, r) \in \mathbb{N}$ such that, for any $n \geq n_0$, $P^{(n)}_{m,\epsilon}(\mathcal{X}) \geq 1 - \delta$ in the random experiment $(\Omega^{(n)}_{m,\epsilon}, \mathcal{B}^{(n)}_{m,\epsilon}, P^{(n)}_{m,\epsilon})$.*

Proof. The proof follows along the lines of the proof of Theorem 3 in [25]. An outline is given below. For details, see [5].

We make a geometric construction as follows: construct radial "blades" (see Figure 4) to cover the entire portion of the circle of radius $h_{\max} r$ centered at the BS, and lying inside the area A. Then, in each of these blades, we construct $h_{\max}$ rectangular strips (again see Figure 4). Denote by $\mathcal{X}_{\epsilon,\delta}$, the event that at least one potential location (out of the n uniformly distributed locations) falls in each of the first $(h_{\max} - 1)$ strips (counting from the BS at $(0, 0)$) in each blade. We can choose the widths and separation of the strips such that occurrence of the event $\mathcal{X}_{\epsilon,\delta}$ ensures that the resulting instance has a feasible solution, i.e., $\mathcal{X}_{\epsilon,\delta} \subseteq \mathcal{X}$, or equivalently, $P^{(n)}_{m,\epsilon}(\mathcal{X}) \geq P^{(n)}_{m,\epsilon}(\mathcal{X}_{\epsilon,\delta})$. Then, we use the union bound along with some geometric arguments to lower bound $P^{(n)}_{m,\epsilon}(\mathcal{X}_{\epsilon,\delta})$ as a function of n, $h_{\max}$ and ϵ. This, in conjunction with the requirement that $P^{(n)}_{m,\epsilon}(\mathcal{X}) \geq 1 - \delta$, yields $n_0(\epsilon, \delta, h_{\max}, r)$, and the sufficient condition that $n \geq n_0$ ensures this probability. □

Remark: For fixed $h_{\max}$ and r, $n_0(\epsilon, \delta)$ increases with decreasing ϵ and δ.

The experiment: In light of Theorem 2, we employ the following node deployment strategy to ensure, w.h.p, feasibility of the RST-MR-HC problem in the area A. Choose arbitrary small values of $\epsilon, \delta \in (0, 1)$. Given the hop count bound $h_{\max}$ and the maximum communication range r, obtain $n_0(\epsilon, \delta, h_{\max}, r)$ as defined in Theorem 2. Deploy $n \geq n_0$ potential locations i.i.d uniformly randomly over the area of interest, A. m sources are deployed i.i.d uniformly randomly within a radius $(1-\epsilon)h_{\max} r$ from the BS, i.e., over the area A_ϵ. By virtue of Theorem 2, this ensures that any source deployed within a distance $(1-\epsilon)h_{\max} r$ is no more than $h_{\max}$ hops away from the BS w.h.p, thus ensuring feasibility of the RST-MR-HC problem w.h.p. We check whether the deployment is feasible by computing the SPT on the induced random geometric graph with hop count as cost. *In this stochastic setting, we derive an upper bound on the average case approximation ratio, α, of the SPTiRP algorithm* as follows.

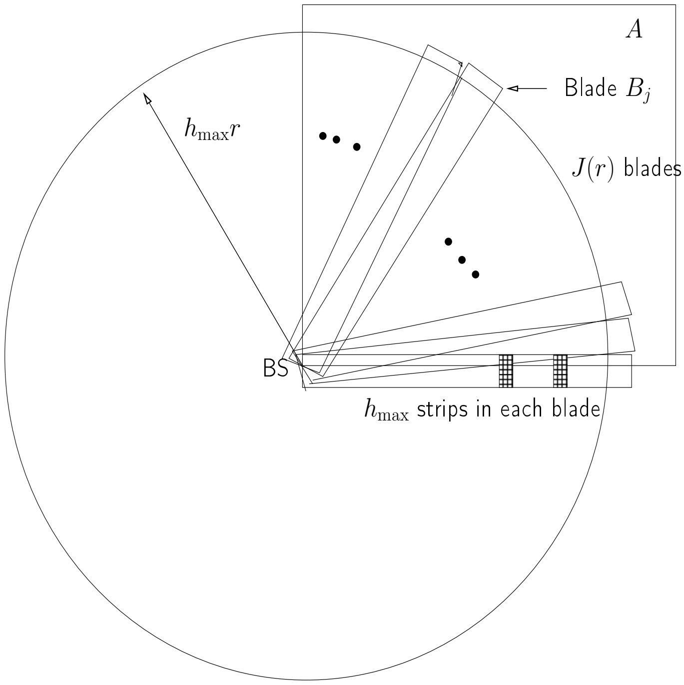

Figure 4: Construction using the blades cutting the circumference of the circle of radius $h_{\max} r$ (adapted from Nath et al. [25]).

Lemma 1.

$$E[N_{\text{SPTiRP}}|\mathcal{X}] \leq m\left[h_{\max} - \frac{1}{(1-\epsilon)^2 h_{\max}^2} - \sum_{j=2}^{h_{\max}-1} \frac{j^2}{h_{\max}^2}\right] - m + \ m\delta(h_{\max} - 1) \tag{2}$$

Proof. We provide an outline below. For details, see [5].

We upper bound the desired conditional expectation by the conditional expectation (conditioned on $\mathcal{X}$) of the relay count in an SPT (since the SPTiRP algorithm starts by finding an SPT, and prunes relays therefrom), which in turn, is upper bounded as follows:

1. Conditioning on the event $\mathcal{X}_{\epsilon,\delta}$ defined in the earlier proof, and upper bounding the resulting conditional expectation using geometric arguments, and the fact that the potential locations are uniformly distributed.
2. Observing that for a deployment in $\mathcal{X} \backslash \mathcal{X}_{\epsilon,\delta}$, the number of nodes in the SPT can be trivially upper bounded as $m(h_{\max} - 1)$.

Combining the above upper bounds yields the desired result. □

Lemma 2.

$$E[R_{\text{Opt}}|\mathcal{X}] \geq \left[1 - \left(\frac{h_{\max} - 1}{(1-\epsilon)h_{\max}}\right)^{2m}\right](1-\delta) \sum_{i=1}^{h_{\max}-1} \left(1 - \frac{\frac{n_i^2}{3}}{(1-\epsilon)^2 h_{\max}^2}\right)^{m-1} \tag{3}$$

where, $n_i = \min(i, h_{\max} - i)$*.*

Proof. We provide an outline below. For details, see [5].

We consider the joint event that $\mathcal{X}_{\epsilon,\delta}$ has occurred, *and* the farthest source is beyond $(h_{\max} - 1)r$ from the BS. Denote this event by $\mathcal{X}_{h_{\max}}$. Then, we can make the following observations:

1. For any instance $\omega \in \mathcal{X}_{h_{\max}}$, the optimum relay count, $R_{\text{Opt}}(\omega)$, is *lower bounded by the number of relays in the path from the farthest source,* $s(\omega)$*, to the BS in any optimal solution in* ω*.*
2. *Irrespective of which optimal solution we consider in* ω, the j^{th} intermediate node in the path from $s(\omega)$ to the BS is located in the lens shaped intersection of the circle of radius jr centered at $s(\omega)$, and the circle of radius $(h_{\max} - j)r$ centered at the BS (see Figure 5). Further, these lenses are disjoint.

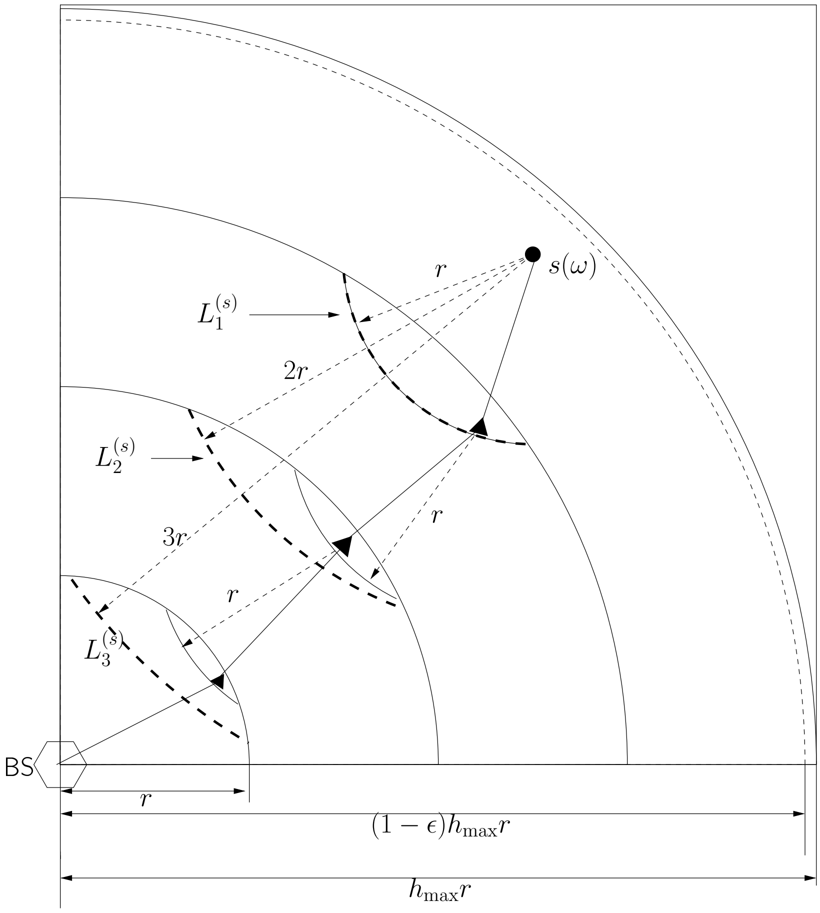

Figure 5: Illustration of the lenses $L_j^{(s)}$, $1 \leq j \leq h_{\max}-1$, used in the proof of Lemma 2; $L_j^{(s)}$ contains in it, the j^{th} lens, and hence the j^{th} intermediate node in a feasible path from source $s(\omega)$ to the BS. The solid triangles indicate the intermediate nodes in a feasible path from source $s(\omega)$ to the BS.

This enables us to define random variables (uniquely determined by ω), to lower bound $R_{\text{Opt}}(\omega)$ (conditioned on $\mathcal{X}_{h_{\max}}$). We can then show, using some probabilistic and trigonometric arguments, that the conditional expectation $E[R_{\text{Opt}}|\mathcal{X}_{h_{\max}}]$ is lower bounded by the third term in the R.H.S of Lemma 2. Finally, the desired Lemma follows by using certain conditional independence arguments to lower bound $E[R_{\text{Opt}}|\mathcal{X}]$ in terms of $E[R_{\text{Opt}}|\mathcal{X}_{h_{\max}}]$. □

It follows from Lemma 1 and Lemma 2 that:

Theorem 3. *The average case approximation ratio of the SPTiRP algorithm over all feasible instances in the stochastic setting described earlier is upper bounded as*

$$\alpha \quad \leq \frac{\overline{N}}{\underline{R}_{\text{Opt}}} \tag{4}$$

where, $\overline{N}$ is given by the R.H.S of (2), and $\underline{R}_{\text{Opt}}$ is given by the R.H.S of (3).

Related numerical experiments are presented in Section 5.1.

5. SPTiRP: Numerical Results

We performed two sets of experiments to test the SPTiRP algorithm. In all these experiments, the relays and the sources are placed randomly. The first set of experiments were performed with a large number of relays, in a setting that conforms to the conditions mentioned in Theorem 2, and hence a feasible solution is guaranteed with high probability. However, due to the large number of relays only a lower bound to the optimum value can be obtained. The second set of experiments were performed with a small number of relays, so that feasibility cannot be assured, but the optimum value can be obtained in every feasible instance.

In Experiment Set 1, we need a large number of potential relay locations to ensure the high probability of feasibility. For such large problem instances, an exhaustive enumeration of all possible solutions to obtain the optimum solution is impractical. Hence, for each of these problem instances, we obtained a *lower bound* on the optimum relay count by solving the LP relaxation of an ILP formulation for the RST-MR-HC problem. For details of the ILP formulation, see [5].

In Experiment Set 2, however, the number of potential relay locations, and hence, the problem size was moderate; so we obtained the *exact* optimum relay count for each instance by an exhaustive enumeration technique, starting with the solution provided by the SPTiRP algorithm. The details are provided below.

5.1. Experiment Set 1

We generated 100 random networks as follows (in what follows, all distances are in meters, unless otherwise mentioned): we chose $r = 60$, and $h_{\max} = 4$ for this set of experiments. We also chose $\epsilon = \delta = 0.1$ (see Theorem 2). For the chosen parameter values and for a square area of 216×216, the required number of potential relay locations was found to be $n(\epsilon, \delta, h_{\max}, r) \geq 1908$. Hence, 1908 potential relay locations were selected uniformly randomly over a 216×216 square area. This ensures that any point within a distance $(1 - \epsilon)h_{\max}r$ from the BS is at most $h_{\max}$ hops away from the BS with a high probability ($\geq (1 - \delta) = 0.9$). 10 source nodes were deployed uniformly randomly over the quarter circle of radius $(1 - \epsilon)h_{\max}r = 216$; hence we have a feasible solution with a high probability (≥ 0.9).

The SPTiRP algorithm was run on the 100 scenarios thus generated; none of the 100 scenarios tested turned out to be infeasible. For each scenario, a lower bound on the optimum relay count was obtained by solving the LP relaxation of the corresponding ILP formulation.

The results are summarized in Table 1.

Table 1: Test Set 1: Performance of the SPTiRP algorithm compared to a lower bound on the optimum solution

Potential relay count	Scenarios	Design matched with lower bound (hence, was optimal)	Off by one from lower bound	Max off from lower bound
1908	100	23	21	10

Observations

1. In 44% of the tested scenarios, the algorithm ends up giving optimal or near-optimal (exceeding the optimum by just one relay) solutions. However, note that the comparison was only against a lower bound on the optimum solution, which can potentially be loose depending on the problem scenario, and we suspect the actual performance of the algorithm to be much better (indeed, as we shall see in Experiment Set 2 by comparing against the actual optimum solution, the algorithm performed close to the optimum in most of the tested scenarios).
2. In the remaining cases, where it is off by more than one relay, the maximum difference from the lower bound was found to be 10 relays.
3. We computed the empirical worst case approximation factor from the experiments as follows: for each scenario, we computed the approximation factor given by the SPTiRP algorithm w.r.t the lower bound obtained from the LP relaxation as *approximation factor* $= \frac{\text{Relay}_{\text{Algo}}}{\text{Relay}_{\text{lowerbound}}}$. The maximum of these over all the tested scenarios (in the current set of experiments) was taken to be the (empirical) worst case approximation factor.
4. We also computed the theoretical bound on the average approximation ratio for the given setting and parameter values using Equation 4, and compared it against the empirical average case approximation ratio obtained from the experiments as

$$\text{Empirical average case approx. ratio} = \frac{\text{Average relaycount of SPTiRP over 100 scenarios}}{\text{Average lower bound from LP relaxation over 100 scenarios}} \tag{5}$$

The results are summarized in Table 2.

Table 2: Test Set 1: Approximation ratio for the SPTiRP algorithm

Potential relay count	Scenarios	Worst case approximation ratio		Average case approximation ratio	
		Theoretical	Experimental	Theoretical bound (Eqn. (4))	Experimental(Eqn. (5))
1908	100	30	5	14	1.66

In Table 3, we have compared the execution time of the SPTiRP algorithm against the time required to compute a lower bound on the optimum solution by solving the LP relaxation. Both the algorithms were run in MATLAB 7.11, using a single compute node (Linux based) with 16 GB main memory, and a single processor with 4 cores, i.e., 4 CPUs. As can be seen from the table, while the SPTiRP algorithm computes a very good (often optimal) solution in at most a few seconds, computing even the lower bound on the optimal solution (i.e., solving the LP relaxation instead of the actual ILP) can be actually quite time consuming, running into several hours (up to about 12 hours in the worst case).

Table 3: Test Set 1: Computation time of the SPTiRP algorithm compared to optimal solution (lower bound) computation

Potential Relay Count	Scenarios	Mean execution time of SPTiRP in sec	Mean Execution time of obtaining a lower bound on optimal solution in sec	Max execution time of SPTiRP† in sec	Max execution time of obtaining a lower bound on optimal solution† in sec
1908	100	6.6621	7002	18.4438	41716

† These were *not* attained on the same problem

5.2. *Experiment Set 2*

In this set of experiments we deployed a smaller number of relays randomly. Due to the small number of relays, the probabilistic analysis of feasibility is not useful. We generated 1000 random networks as follows (in what follows, all distances are in meters, unless otherwise mentioned): A 150×150 square area is partitioned into square cells of side length 10. Consider the lattice created by the corner points of the cells. 10 source nodes are placed at random over these lattice points. Then the potential relay locations are obtained by selecting n points uniformly randomly over the 150×150 area; n was varied from 100 to 140 in steps of 10, and for each value of n, we generated 200 random network scenarios (thus yielding 1000 test cases). We chose $r = 60$, and $h_{\max} = 6$ for the experiments.

Given the outcome of the SPTiRP algorithm, an optimal solution can be obtained as follows: Suppose the SPTiRP algorithm uses n relays. Then perform an exhaustive search over all possible combinations of $(n - 1)$ and fewer relays to check if the hop constraint can still be met.

In none of the 1000 scenarios tested, the hop constraint turned out to be infeasible. The results are summarized in Table 4.

Table 4: Test Set 2: Efficiency of the SPTiRP algorithm in obtaining the optimal design

Potential relay count	Scenarios	Optimal design	Off by one from optimal	Max off from optimal
100	200	154	42	3
110	200	154	40	2
120	200	158	39	2
130	200	155	36	2
140	200	161	38	2
Total	1000	782	195	3

Observations

1. As in the case of test set 1, even for test set 2, in over 97% of the tested scenarios, the algorithm ends up giving optimal or near-optimal (exceeding the optimum by just one relay) solutions.
2. In the remaining cases, where it is off by more than one relay, the maximum difference was found to be 3 relays.

In Table 5, we have compared the execution time of the SPTiRP algorithm against the time required to compute an optimal solution, given the outcome of the SPTiRP algorithm. Both the SPTiRP algorithm, and the postprocessing on its outcome were run in MATLAB 7.0.1 on a Windows Vista (basic) based PC having Intel Core 2 Duo T5800 CPU with processor speed of 2 GHz, and 3 GB RAM. Again, while the SPTiRP algorithm computed a very good (often optimal) solution in at most a second or two (averaging less than a second), computing the optimum solution even after being provided with a very good upper bound on the required number of relays by SPTiRP, turned out to be quite time consuming, running into several minutes.

Also, we note from Table 5 that, as the node density increases, the computation time of the SPTiRP algorithm also increases.

Remark: Both the above sets of experiments were performed on geometric graphs. However, our algorithm is applicable to any arbitrary input graph. Hence, to test the performance of our algorithm on non-geometric input graphs,

Table 5: Test Set 2: Computation time of the SPTiRP algorithm compared to optimal solution computation

Potential Relay Count	Scenarios	Mean execution time of SPTiRP in sec	Mean execution time of directly obtaining an optimal solution in sec	Max execution time of SPTiRP† in sec	Max execution time of directly obtaining an optimal solution† in sec
100	200	0.588	661.485	1.638	1828.7
110	200	0.705	240.85	2.081	722.29
120	200	0.812	423.89	1.591	944.74
130	200	0.993	951.495	2.606	2674.9
140	200	1.144	140.7	2.808	355.46
Overall	1000	0.848	483.684	2.808	2674.9

† These were not necessarily attained on the same problems

we have conducted experiments with Erdos-Renyi random graphs, where the feasible edges in the graph are chosen i.i.d with some probability p. The results are extremely encouraging; in all the experiments conducted, the algorithm ended up with an optimal solution. We omit the details of the experiments here. See [5] for details.

6. SPTiRP: Simulation Results

To test the QoS under positive traffic arrival rates, of the network topologies obtained using SPTiRP algorithm, we performed extensive simulations using Qualnet v4.5 [26]. For these simulations, we assumed the PHY and MAC layers to be as specified in the IEEE 802.15.4 standard[27].

We generated 20 network topologies as follows: in each case, 10 source nodes, and 120 potential relay locations were randomly selected in a 150×150 square meters area in exactly the same way as described in Section 5.2. As before, the BS was assumed to be at the corner $(0, 0)$. We chose the **maximum communication range,** $r = 30$ **meters**, which, for a **transmit power of 0 dBm**, and a **PHY layer packet size of 131 bytes**, corresponds to a **PER of** $\leq 1\%$, assuming the path loss model given in the standard [27, 28], a **fade margin of 20 dB**, and **receiver sensitivity of** -98.8 **dBm**. The *hop constraint was chosen as* $h_{\max} = 9$, which, for a PER of 1%, corresponds to an *end-to-end delivery probability of 91.35% (under the lone packet model)*, and an *end-to-end mean delay of 56.16 msec, also under the lone packet model*, assuming the CSMA/CA backoff parameters given in the standard, and a PHY layer packet size of 131 bytes (see [11] for details of how this mean end-to-end delay can be computed). Having chosen r, we had a graph on the sources, and the potential relay locations. We used the SPTiRP algorithm on this network graph with the above mentioned hop constraint, to obtain a tree topology connecting the sources to the BS using a small number of relays, and satisfying the hop constraint.

Qualnet simulation was performed on each of the 20 network topologies thus generated, for **six different traffic arrival rates**, namely, λ =0.1, 0.2, 0.3, 0.4, 0.5, and 2 packets/sec from each source. The *arrival process was assumed to be Poisson*. The simulation procedure is described below:

1. We used the following *interference model* in Qualnet: any two nodes that are within *Carrier Sense* (CS) range of each other can hear each other's transmission. If two nodes are within the CS range of a receiver node, then their transmissions interfere with each other at the receiver node. The CS range, r_{cs}, was set equal to r for the simulations (see above).
2. We used the *collision model* in Qualnet to account for packet losses due to interference. If two or more packet transmissions interfere with one another at a receiver node, then all of those packets are lost.
3. For each topology, and each arrival rate, *the simulation was repeated for 25 iterations, with each iteration being run for 1500 seconds of simulated time..*
4. For each topology, and each arrival rate, we recorded the end-to-end delivery probability (we shall use the shorthand p_{del} for this from now on, with slight abuse of notation) from each source to the sink, averaged over the 25 iterations, and the mean end-to-end packet delay from each source to the sink, also averaged over 25 iterations.
   The results are summarized in Table 6. To keep the table concise, we have adopted the following strategy: for each arrival rate and each topology, we have computed p_{del} *averaged over the 10 sources*, and reported only the minimum average p_{del} for each rate, the minimum being taken over the 20 scenarios. This constitutes column 3 of Table 6. A similar strategy has been adopted for reporting the end-to-end delay (column 6 of the table). We have also reported the minimum p_{del} and the maximum delay *encountered over all sources and all the 20 scenarios* for each rate (columns 2 and 5 respectively), and the maximum p_{del} and the minimum delay *encountered over all sources and scenarios* for each rate (columns 4 and 7 respectively).

Observations:

Table 6: Summary of Qualnet simulation results for 20 network topologies

Arrival rate in pkts/sec	Minimum p_{del}	Minimum average p_{del}	Maximum p_{del}	Maximum delay in sec	Maximum average delay in sec	Minimum delay in sec
0.1	0.874	0.905	0.982	0.0509	0.0401	0.0113
0.2	0.860	0.893	0.980	0.0510	0.0401	0.0113
0.3	0.846	0.879	0.981	0.0511	0.0402	0.0113
0.4	0.826	0.865	0.980	0.0513	0.0403	0.0114
0.5	0.802	0.848	0.978	0.0514	0.0404	0.0114
2.0	0.557	0.667	0.967	0.0535	0.0417	0.0117

1. From Table 6, we observe that the mean end-to-end delay *never exceeded* the lone-packet target end-to-end delay of 56.16 msec.
2. For low arrival rates, the minimum p_{del} violated the lone-packet target p_{del} only by a small margin. The maximum percentage violations in p_{del} w.r.t the lone packet target p_{del} are summarized in Table 7.

Table 7: Proximity of positive traffic QoS to lone-packet target QoS

Arrival rate in pkts/sec	Maximum percent violation in p_{del} w.r.t lone-packet target (over the 20 scenarios tested)
0.1	4.37
0.2	5.88
0.3	7.37
0.4	9.59
0.5	12.17
2.0	39

3. From Table 7, we note that the tested network topologies, although designed for the lone-packet model, *can handle light positive traffic arrival rate (up to 0.4 packets/sec) from each source, without exceeding the lone-packet QoS by more than 10%.*

7. Related Literature

The problem we have chosen to address belongs, broadly, to the class of Steiner Tree Problems (STP) on graphs ([29, 30]).

The classical STP is stated as: *given an undirected graph $G = (V, E)$, with a non-negative weight associated with each edge, and a set of* required vertices *$Q \subseteq V$, find a minimum total edge cost subgraph of G that spans Q, and may include vertices from the set $S := V - Q$, called the* Steiner vertices.

The classical STP dates back to Gauss and it has been proven to be NP-Hard. Lin and Xue [20] proposed the Steiner Tree Problem with Minimum Number of Steiner Points and Bounded Edge Length (STP-MSPBEL). The STP-MSPBEL was stated as: given a set of n terminal points Q in 2-dimensional Euclidean plane, find a tree spanning Q, and some additional Steiner points such that each edge has length no more than R, and the number of Steiner points is minimized. This bound on edge length only constrains link quality, but not end to end QoS. The problem was shown to be NP-complete and a polynomial time 5-approximation algorithm was presented. This problem was the first well-studied problem on optimal relay placement (the relay locations were *unconstrained*). However, no average case performance guarantee was provided for the proposed algorithm.

Cheng et al. [31] studied the same problem as Lin and Xue, and proposed a 3-approximation algorithm and a 2.5-approximation algorithm.

Lloyd and Xue [21] studied a generalization of STP-MSPBEL problem where each sensor node has range r and each relay node has range $R \geq r$. They provided a 7-approximation polynomial time algorithm. They also studied the problem of minimum number of relay placement such that there exists a path consisting solely of relay nodes between each pair of sensors. For this problem, they provided a $(5 + \epsilon)$-approximation algorithm. The problems studied by Lloyd and Xue, as well as Cheng et al. fall in the category of *unconstrained* relay placement problem. Neither work provide any average case performance guarantee of their proposed algorithms.

Hao et al.[22], and Zhang et al.[23] extended the problem studied by Lloyd and Xue [21] to incorporate *fault tolerance* by ensuring two-connectivity while deploying a minimum number of relay nodes. They provided polynomial time approximation algorithms for the problems. In their formulations, the relay node locations are *unconstrained*.

Moreover, they did not consider any hop constraint in their formulation. Also, no average case analysis was provided for the proposed algorithms.

Voss [10] studied the Steiner Tree Problem with Hop Constraints (STPH). This problem is stated as: *given a directed connected graph $G = (V, E)$, with non-negative weight associated with each edge, consider a subset of V, namely, $Q = \{0, 1, 2, \dots, m\}$ with 0 being the root vertex, and a positive integer H. The problem is to find a minimum total edge cost subgraph T of G such that there exists a path in T from 0 to each vertex in $Q\backslash\{0\}$ not exceeding H arcs (possibly including vertices from $S := V - Q$).* We can call this problem the *Rooted Steiner Tree-Minimum Weight-Hop Constraint* problem (RST-MW-HC). This problem was shown to be NP-Hard, and a Minimal Spanning Tree based heuristic algorithm was proposed to obtain a good quality feasible solution, followed by an improvement procedure using a variation of *Local Search* method called the *Tabu search* heuristic. *No performance guarantee or complexity analysis of the heuristic was provided.* Also, the tabu search heuristic may not be polynomial time.

Note that an instance of the RST-MR-HC problem can be converted to an instance of the RST-MW-HC problem in polynomial time as follows: replace each relay with a directed edge of weight 1, and replace each edge associated with the relay with two directed edges (each of weight 0), one incident into the tail of the edge substituting the relay, and one going out of the tip of the edge substituting the relay. Then, minimizing the number of relays in the original problem is equivalent to minimizing the total weight in the converted problem. Then, one could use Voss's algorithm on this instance of RST-MW-HC problem to solve the original problem. But, as we mentioned earlier, Voss's algorithm does not provide any performance guarantee, and because of the tabu search heuristic (which may not be polynomial time), it may take long to converge to a solution.

Costa et al. [32] studied the Steiner Tree Problem with revenue, budget, and hop constraints. Given a graph $G = (V, E)$, with a cost associated with each edge, and a *non-negative revenue* associated with each vertex, the problem is to determine a revenue maximizing tree subject to a total edge cost constraint, and a hop constraint between the root vertex and every other vertex in the tree. They propose a greedy algorithm for initial solution followed by destroy-and-repair or tabu search to improve the initial solution. They have evaluated the performance of the proposed algorithms only through numerical experiments; *no theoretical guarantee has been provided.*

It is possible to cast our problem into the form of the one addressed by Costa et al. [32] as follows: assign a *negative* revenue, say -1, to each relay node (Steiner vertex), and a large positive revenue, say $|R| + 1$, where $|R|$ is the number of potential relay locations, to each source vertex. This cost assignment would ensure that a revenue maximizing tree has all the source vertices in it, since the gain in revenue by adding a source outweighs the loss in revenue due to the additional relays, if any, required to connect the source to the BS. Also, the negative revenue on relays ensures that the revenue maximizing tree contains in it, as few relays as possible. Now, choose the hop constraint to be the same as that in the original RST-MR-HC problem. Also, assign a cost of zero to each edge, and choose a trivial total edge cost constraint (any positive real number). With these assignments/choices, the problem of minimizing total relay count while obtaining a hop constrained tree network (RST-MR-HC) is the same as the problem of obtaining a revenue-maximizing Steiner tree subject to a hop constraint and a total edge cost constraint. This formulation, however, requires the node weights to be negative, whereas the algorithm proposed by Costa et al. *requires* the non-negativity of the node weights [5]. Moreover, even if one could find a way to map the RST-MR-HC problem to the revenue-budget-hop constrained STP, the tabu search based heuristic proposed by Costa et al. to improve the initial solution to the revenue-budget-hop constraint problem is not guaranteed to be polynomial time in general, and may take a long time to converge.

Kim et al. [33] studied the Delay and Delay Variation Constrained multicasting Steiner Tree problem. The problem is similar to the one studied by Voss, with a delay constraint instead of the hop constraint, and a constraint on delay variation between two sources. With the delay variation constraint relaxed, Kim's problem becomes the *Rooted Steiner Tree-Minimum Weight-Delay Constraint* problem. They proposed a polynomial time heuristic algorithm to obtain feasible solutions, but they also did not provide any performance guarantee for their algorithm.

Bredin et al. [34] studied the problem of optimal relay placement *(unconstrained)* for $k-$connectivity. They proposed an $O(1)$ approximation algorithm for the problem with any *fixed* $k \geq 1$. However, they did not provide any average case analysis for their algorithm.

Klein and Ravi studied the *constrained* relay placement problem for connectivity in *arbitrary* input graphs (not necessarily geometric); they called this problem, the *node-weighted* Steiner tree problem. Using our nomenclature, we can also call this problem, the *Rooted Steiner Tree-Minimum Relays* (RST-MR) problem. They proved that the best possible approximation factor for this problem is $O(\log m)$, and provided an algorithm that achieves this guarantee.

[5]The greedy algorithm that they proposed starts with the root node, and proceeds by adding a path connecting a non-selected *profitable* vertex to the existing solution at each step; when the budget constraint can be trivially satisfied, this amounts to simply finding a hop constrained path from a profitable vertex to the root. This is not enough to ensure revenue maximization if the revenues associated with some of the nodes is negative, since the path selected from the profitable vertex to the root may contain vertices with negative revenue, thus reducing the profit along the way; thus, additional constraints must be imposed for selection of paths from the profitable vertices to the root node to ensure minimal usage of the negative-revenue vertices.

However, they did not consider any hop constraint or diameter constraint in their formulation.

Misra et al. [35] studied the *constrained* relay placement problem for connectivity and survivability for the special case of geometric graphs where the feasible edges are defined in terms of a maximum allowed edge length. They provided $O(1)$ approximation algorithms for both the problems. While the edge length bound in their formulation can model the link quality, the formulation does not involve a path constraint such as the hop count along the path; hence, there is *no constraint on the end-to-end QoS.*

Yang et al. [36] studied a variation of the problem in [35], namely the *two-tiered* constrained relay placement problem for connectivity and survivability, where each source has to be *covered* by one (two) relay nodes, and the relay nodes form a one (two)-connected network with the BS. They provided $O(\ln m)$ approximation algorithms for arbitrary settings, and $O(1)$ approximation for some special cases. Their formulation also *does not involve any constraint on the end-to-end QoS*.

The numerical experiments in both [35] and [36] actually evaluate the empirical average case performance of their proposed algorithms on random test scenarios, which they compare against the theoretically derived *worst case* performance bounds. Neither work, however, attempt a formal analysis of the average case performance of the proposed algorithms.

Sitanayah et al. [7, 8] addressed a more general version of the RST-MR-HC problem where they considered $k-$connectivity with $k \geq 1$. We call this problem, the Rooted Steiner Network for k-connectivity with Minimum Relays and Hop Constraint (RSNk-MR-HC) problem. They proposed local search based heuristics for the problem. However, they *did not provide any analysis of the complexity of the problem, or the performance guarantee (time complexity, worst case and average case approximation guarantees) of their algorithms*. Moreover, in their numerical experiments, they only compared their algorithm against a suitably modified version of another existing heuristic, and *no comparison against the optimal solution has been provided*. Note that we have also developed algorithms for the RSNk-MR-HC problem, and analyzed the performance of our proposed algorithms. Owing to lack of space, we do not include that work here. See [5] for details.

Recently, Nigam and Agarwal [9] have proposed a branch-and-cut algorithm to solve the RST-MR-HC problem optimally. However, unlike the SPTiRP algorithm, *their algorithm can be used only for a subclass of problems where none of the source nodes have a singleton node cut, i.e., deletion of a single node does not cause the hop constraint to be infeasible for any source*. To emphasize this fact, we refer to their problem as the RST-MR-HC-Subclass problem. Moreover, their algorithm is *exponential time* in general, and hence cannot be used for large sized problem instances such as the ones described in Section 5.1. Even for problem instances of moderate size comparable to our test instances in Section 5.2, the running time of their algorithm was of the order of minutes in the worst case [6](see Table 3 in [9]) whereas the SPTiRP algorithm has running time of the order of one or two seconds while still being near-optimal in 97% of the tested scenarios.

Another class of problems arising in the context of telecommunication networks is the bounded diameter minimum spanning tree (BDMST) problem. Given a graph $G = (V, E)$, with edge cost $c(e) \geq 0$ associated with each edge $e \in E$, this problem asks for a tree spanning the vertex set V, with diameter not exceeding a given integer, D, and having the minimum total edge cost. Note that *this is not a relay placement problem*, since there are no Steiner vertices to be chosen. *All the vertices in the input graph G must be part of the final solution*. Gruber and Raidl [37] studied the *Euclidean* version of the problem, and proposed a meta-heuristic for the problem. They did not provide any performance guarantee for their algorithm. Also, the resulting solutions were compared only against existing heuristics; no comparison with optimal solutions were provided. Moreover, recall from Proposition 2 that the BDMST problem is a special case of the diameter bounded node-weighted Steiner tree problem. Thus, clearly, we cannot hope to solve the general problem by only using an algorithm for the special case.

Table 8: A comparison with closely related literature; the "starred" problem is the one we address in this paper (we have also addressed the problem of k-connectivity where $k > 1$. See [5] for details); an entry '×' in a column means that the corresponding algorithm does not provide the attribute given in the top of that column, whereas a '✓' means that it does provide the attribute.

Problem	End-to-end performance objective	Complexity	Time complexity guarantee of proposed algorithm	Worst case approximation guarantee of proposed algorithm	Average case approximation guarantee of proposed algorithm
RST-MR [35]	×	NP-Hard	polynomial time	6.2	×
RST-MW-HC [10]	✓	NP-Hard	×	×	×
RST-MW-DC [33]	✓	NP-Hard	polynomial time	×	×
RSNk-MR-HC [7, 8]	✓	NP-Hard	×	×	×
RST-MR-HC-Subclass [9]	✓	NP-Hard	Exponential time	1	1
RST-MR-HC*	✓	NP-Hard	polynomial time	polynomial factor	polynomial factor

In Table 8, we present a brief comparison of the problem under study in this paper with some of the closely related

[6]Despite using more advanced processor, and computational software than ours, although with somewhat less RAM space.

problems studied in the literature.

8. Conclusion

In this paper, we have studied the problem of determining an optimal relay node placement strategy such that certain performance objective(s) (in this case, hop constraint, which, under a lone-packet model, ensures data delivery to the BS within a certain maximum delay) is (are) met. We found that the problem is NP-Hard, and proposed polynomial time approximation algorithm for the problem. The algorithm is simple, intuitive, and, as can be concluded from numerical experiments presented in Section 5, gives solutions of very good quality from a practical perspective, using extremely reasonable computation time. We have also provided worst case and average case bounds on the performance of the algorithm. The average case analysis technique, while yielding loose bound compared to numerical experiments, may still be of independent interest since it seems to be the first of its kind in the relay placement literature.

One might ask why the local search algorithms presented in this paper work so well in the tested random scenarios. The answer to this question is not immediately obvious, but, for the RST-MR-HC problem, the graphs we ran our tests on were all geometric graphs; hence, a formal analysis of the properties of the underlying random geometric graph might provide some useful insights into the performance of these local search algorithms. We wish to address this issue in our future work.

Further, we are working on extending the design to traffic models more complex than the lone packet traffic model considered here. This requires the analysis of packet delays in a mesh network with more complex traffic flows and the nodes accessing the medium using CSMA/CA as defined in IEEE 802.15.4 (see [38] and [15] for efforts in this direction).

References

[1] Honeywell, "www.honeywell.com/ps/wireless."

[2] ISA100, "www.isa.org/isa100."

[3] GE, "http://www.ge.com/stories/industrial-internet."

[4] R. A. Sajana, R. Subramanian, P. V. Kumar, S. Krishnan, B. Amrutur, J. Sebastian, M. Hegde, and S. Anand, "A low-complexity algorithm for intrusion detection using a PIR-based wireless sensor network," in *International Conference Series on Intelligent Sensors, Sensor Networks and Information Processing*, (Sydney), 2009.

[5] A. Bhattacharya and A. Kumar, "QoS aware and survivable network design for planned wireless sensor networks," tech. rep., available at arxiv.org/pdf/1110.4746, 2013.

[6] Z. Alliance, "Zigbee document 02130r9: Network specification, revision 9," July 2004.

[7] L. Sitanayah, K. N. Brown, and C. J. Sreenan, "Fault-tolerant relay deployment for k node-disjoint paths in wireless sensor networks," in *4th IFIP Wireless Days International Conference*, 2011.

[8] L. Sitanayah, K. N. Brown, and C. J. Sreenan, "Fault-tolerant relay deployment based on length-constrained connectivity and rerouting centrality in wireless sensor networks," in *9th European Conference on Wireless Sensor Networks (EWSN)*, pp. 115–130, 2012.

[9] A. Nigam and Y. K. Agarwal, "Optimal relay node placement in delay constrained wireless sensor network design," *European Journal of Operational Research*, vol. 233, pp. 220–233, 2014.

[10] S. Voss, "The Steiner tree problem with hop constraints," *Annals of Operations Research*, vol. 86, pp. 321–345, 1999.

[11] A. Bhattacharya, "Node placement and topology design for planned wireless sensor networks," M.E thesis, Indian Institute of Science, June 2010.

[12] Cyan, "http://www.cyantechnology.com/apps/index.php."

[13] B. Aghaei, "Using wireless sensor network in water, electricity and gas industry," in *3rd IEEE International Conference on Electronics Computer Technology*, pp. 14–17, April 2011.

[14] S. Borst, M. Jonckheere, and L. Leskela, "Stability of parallel queueing systems with coupled service rates," *Discrete Event Dynamic Systems*, vol. 18, pp. 447–472, 2008.

[15] R. Srivastava and A. Kumar, "Performance analysis of beacon-less IEEE 802.15.4 multi-hop networks," in *4th International Conference on Communication Systems and Networks (COMSNETS)*, 2012.

[16] A. Bhattacharya, S. M. Ladwa, R. Srivastava, A. Mallya, A. Rao, D. G. R. Sahib, S. Anand, and A. Kumar, "Smartconnect: A system for the design and deployment of wireless sensor networks," in *5th International Conference on Communication Systems and Networks (COMSNETS)*, 2013, available at `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6465582`.

[17] P. N. Klein and R. Ravi, "A nearly best-possible approximation algorithm for node-weighted steiner trees," *Journal of Algorithms*, vol. 19, no. 1, pp. 104–115, 1995.

[18] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Bell Telephone Laboratories, Inc., 1979.

[19] D. Panigrahi, *Optimization problems in network connectivity*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2012.

[20] G.-H. 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.

[21] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Transactions on Computers*, vol. 56, January 2007.

[22] B. Hao, J. Tang, and G. Xue, "Fault-tolerant relay node placement in wireless sensor networks: Formulation and approximation," in *HPSR*, 2004.

[23] R. Zhang, T. R. Park, M. J. Lee, H. Jung, and J. Ryu, "Testbed experimentation of a meshed tree routing with local link state for wireless pan mesh," in *ICC*, 2008.

[24] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press and McGraw-Hill, 2 ed., 2001.

[25] S. Nath, N. E. Venkatesan, A. Kumar, and P. V. Kumar, "Theory and algorithms for hop-count-based localization with random geometric graph models of dense sensor networks," *ACM Transactions on Sensor Networks*, pp. 111–150, 2010.

[26] "http://web.scalable-networks.com/content/qualnet."

[27] *IEEE Standards Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, October 2003.

[28] *IEEE Standards Part 15.2: Coexistence of Wireless Personal Area Networks with Other Wireless Devices Operating in Unlicensed Frequency Bands*, August 2003.

[29] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*, vol. 53 of *Annals of Discrete Mathematics*. Elsevier Science Publishers B.V, 1992.

[30] V. V. Vazirani, *Approximation Algorithms*. Springer, 2003.

[31] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Netw*, vol. 14, pp. 347–355, 2008.

[32] A. M. Costa, J.-F. Cordeau, and G. Laporte, "Fast heuristics for the steiner tree problem with revenues, budget and hop constraints," *European Journal of Operational Research*, vol. 190, pp. 68–78, 2008.

[33] M. Kim, Y.-C. Bang, and H. Choo, "On multicasting Steiner trees for delay and delay variation constraints," 2006.

[34] J. L. Bredin, E. D. Demaine, M. T. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance," in *ACM MobiHoc*, 2005.

[35] 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.

[36] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations," *IEEE Transactions on Mobile Computing*, 2011. accepted for publication.

[37] M. Gruber and G. R. Raidl, “Solving the euclidean bounded diameter minimum spanning tree problem by clustering-based (meta-)heuristics,” in *Computer Aided Systems Theory- EUROCAST*, pp. 665–672, 2009.

[38] C. K. Singh, A. Kumar, and P. M. Ameer, “Performance evaluation of an IEEE 802.15.4 sensor network with a star topology,” *Wireless Network*, vol. 14, pp. 543–568, 2008.