Deploying Mesh Nodes under Non-Uniform Propagation

Joshua Robinson*
Narus, Inc.
Sunnyvale, CA

Mohit Singh
Microsoft Research
Cambridge, MA

Ram Swaminathan
HP Labs
Palo Alto, CA

Edward Knightly
Rice University
Houston, TX

***Abstract*—Wireless mesh networks are popular as a cost-effective means to provide broadband connectivity to large user populations. A mesh network placement provides *coverage*, such that each target client location has a link to a deployed mesh node, and *connectivity*, such that each mesh node wirelessly connects directly to a gateway or via intermediate mesh nodes. Prior work on placement assumes wireless propagation to be uniform in all directions, i.e., an unrealistic assumption of circular communication regions. In this paper, we present approximation algorithms to solve the NP-hard mesh node placement problem for non-uniform propagation settings. The first key challenge is incorporating non-uniform propagation, which we address by formulating the problem input as a connectivity graph consisting of discrete target coverage locations and potential mesh node locations. This graph incorporates non-uniform propagation by specifying the estimated signal quality *per link*. Secondly, our algorithms are the first to minimize the number of deployed mesh nodes with constant-factor approximation ratio in the non-uniform propagation setting. To achieve this, we formulate the *Degree-Constrained Terminal Steiner tree problem* and present approximation algorithms which leverage prior results on the Steiner tree problem. Third, it is impractical to measure all possible potential mesh links, and therefore deployment planning must rely on estimations. To address this challenge, we extend our algorithm to iteratively measure the links in the solution Steiner tree, refining the graph input on a per-link basis in order to ensure the deployed network is not disconnected. Finally, we use propagation measurements at 35,000 locations in the deployed GoogleWiFi network to investigate placement in a realistic, non-uniform propagation environment. Under this measured propagation setting, our algorithms result in up to 80% fewer mesh nodes than current algorithms and only require an average of 3 measurements per deployed mesh node to ensure backhaul connectivity.**

I. Introduction

Wireless mesh networks provide broadband Internet access to large contiguous areas through the placement of mesh nodes [9]. Mesh deployment requires selecting the number and locations to place mesh nodes such that the target region is fully covered and the mesh nodes are inter-connected in order to forward traffic to Internet gateway points. Unfortunately, prior placement studies address neither the realistic, outdoor physical-layer environments where propagation is non-uniform nor the case when estimations must be used due to the impracticality of measuring all potential mesh links. This work presents two mesh node placement algorithms: 1) an approximation algorithm to find a placement that is no more than a constant factor larger than the optimal size, and 2) an iterative heuristic to choose a small number of measurements so as to minimize the deployed nodes and guarantee mesh node inter-connectivity.

The first contribution of this paper is to formulate the *mesh node placement (MNP)* problem's input as a general connectivity graph, combining target coverage locations with discrete potential mesh node locations into a single input graph. This is the first MNP formulation to consider the non-uniform propagation scenario by specifying connectivity based on *per-link* estimated signal quality, as opposed to prior work [11], [17], [4] which consider the idealized uniform propagation scenario. In other words, we ensure network coverage using arbitrary coverage regions for each mesh node location, instead of a circular disc. Because of the impracticality of measuring all possible potential links before deployment, we use physical-layer estimation techniques [6], [15] to specify the potential links in the input connectivity graph. Propagation modeling, though, introduces estimation errors, and our formulation allows measurement-driven refinement of the input connectivity graph on a per-link basis, eliminating possible estimation errors on selected links.

The mesh node placement problem is NP-hard and we consequently design polynomial-time approximation algorithms to choose mesh placements, i.e., algorithms with provable bounds on worst-case performance. The first contribution of our algorithms is the use of a Steiner tree framework [3], [10] to *jointly* satisfy client coverage, mesh connectivity, and mesh capacity constraints. Specifically, we present a new problem formulation, termed the *Degree-Constrained Terminal Steiner tree* (DCTST) problem. The DCTST problem selects mesh nodes (i.e. Steiner Points) to build a tree which spans all selected mesh nodes (connectivity) with the constraints of bounded vertex degree (capacity) and the requirement that all target client locations are connected as leafs of the tree (coverage), thereby jointly satisfying mesh network constraints. We then present a DCTST algorithm and prove that it finds a solution tree of weight no more than 3.5 times the optimal. Building on this result, our first approximation algorithm, **Minimize-Nodes**, minimizes the number of deployed mesh nodes with a constant-factor approximation ratio proportional to the capacity bound of the mesh nodes and at most a small constant factor violation of degree bounds. Thus, while prior algorithms [4] also have constant factor approximation ratios, our results represent the first that apply in the non-uniform propagation setting and in the case of known gateway locations. We ensure connectivity with known gateways via adding shortcut edges to the input graph between all pairs of gateways, representing the wired infrastructure's connectivity.

*Work completed while at Rice University. Research supported by HP Laboratories and NSF grants CNS-0721894 and CNS-08102501.

The second algorithm, **Measure-and-Place**, minimizes the number of deployed nodes while also using a small number of measurements to ensure that all selected backhaul links are connected. The key idea of this algorithm is iterative DCTST construction combined by refinement of the input connectivity graph via measurements of selected backhaul links.

Finally, we evaluate the performance of the presented algorithms, comparing with state-of-the-art two-phase algorithms that use geometric disc covering [4]. To consider realistic propagation settings, we use signal strength measurements from 35,000 locations and 150 mesh nodes in the GoogleWiFi network. Our algorithms result in 80% fewer deployed mesh nodes and demonstrate that non-uniform propagation benefits deployment by necessitating half as many deployed mesh nodes as in the uniform propagation setting. Further, we find that with expected levels of physical-layer estimation errors, the Measure-and-Place algorithm requires an average of 2.5 measurements per deployed mesh node in order to guarantee connectivity, i.e., three orders of magnitude fewer measurements than a complete measurement survey.

The rest of this paper is organized as follows. Section II defines the mesh node placement problem. Section III presents new placement algorithms and Section IV then evaluates the proposed placement algorithms. Finally, Section V describes related work, and Section VI summarizes.

II. Placement Formulation

The objective of the mesh node placement problem is to minimize the number of deployed mesh nodes with the constraint of full coverage of the target area and connectivity to the Internet. This section first describes a graph-theoretic specification of potential physical-layer links, used as input to the placement problem. We then describe the three general constraints of coverage, connectivity, and capacity, which an operational wireless mesh network must satisfy.

A. Input Connectivity Graph

We formulate the input to the *mesh node placement* problem as a connectivity graph with nodes corresponding to discrete locations and edges between locations that indicate the existence of usable links. This formulation considers non-uniform propagation settings because the input graph encodes the signal quality of each link independently, as opposed to prior geometric covering approaches which assume one coarse-grained propagation parameter (path loss exponent) for all nodes. More formally, we define the *input connectivity graph* $G = (V, E)$, where both target coverage locations and potential mesh node locations form a unified connectivity graph, as described next.

The nodes in the proposed input graph assume the target area is a discrete set C of target coverage locations. The set C consists of physical coordinates representing target areas where client coverage is desired, analogous to the area to be covered in a geometric formulation. For evaluation purposes, we discretize the target coverage grid to 5 meter spacing, such that no client can be farther from a covered grid point than the accuracy of propagation estimation, and we can include only regions the operator seeks to provide service.

The second aspect of the input vertices is the set of potential mesh node locations, M, which is assumed known. Discrete locations for mesh nodes follows naturally from practical constraints on deployment, such as the availability of lamp posts or other infrastructure for mesh node installation. The vertex set of the input connectivity graph is defined as $V = M \cup C$, the union of potential mesh node locations and coverage locations.

B. Non-Uniform Propagation

The input connectivity graph G consists of the set of links, E, corresponding to the potentially usable set of links. Edges are usable if the estimated or measured signal strength is above a signal strength threshold θ_a for access tier links or θ_b for backhaul tier links.

Specifying each link individually enables us to encode non-uniform propagation. In other words, each potential mesh node location can represent an arbitrary coverage region shape. Figure 1 plots nine examples of measured coverage regions [15], illustrating the degree of non-uniform propagation encountered in practice. The exact physical-layer connectivity, represented as the signal strength on each possible link, is prohibitively expensive to obtain for all pair-wise *potential* mesh node and target coverage locations. Instead, graph G captures realistic propagation behavior by allowing each link to be estimated individually by a state-of-the-art propagation modeling approach [6], [15]. These techniques require a small amount of training measurements in order to use environment information to more accurately predict propagation. This data-driven approach to estimating each possible link's signal quality contrasts with prior work, which estimates one range for all access tier links and one range for all backhaul tier links, i.e., the unrealistic uniform propagation assumption. Note that uniform propagation is a special case of the more general non-uniform formulation.

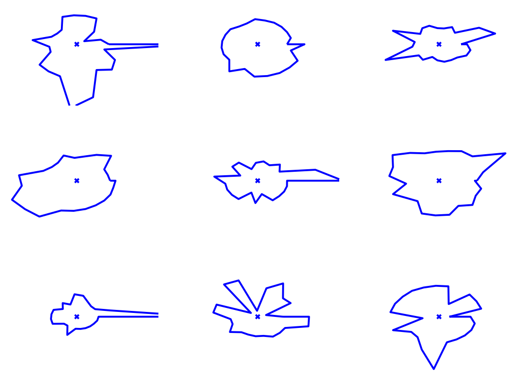

Fig. 1. Nine example coverage regions measured in the GoogleWiFi network in Mountain View, CA, demonstrating non-uniform propagation. Each mesh node location is indicated by 'x's. For scale reference, the top-middle region has average radius of 160 meters.

C. Coverage Constraint

The access tier provides single-hop connectivity from client devices to a mesh node. Correspondingly, the *coverage constraint* requires clients at all target locations in C to be able to connect to at least one mesh node at the specified signal strength threshold θ_a. More formally, let $P \subset M$ be the set of locations selected for mesh node placement. We then require for all locations $c \in C$ that there exists at least one edge (link) in E between location c and one of the mesh nodes in P.

A challenge in formulating the coverage constraint is that it is usually impractical to measure all possible mesh node and coverage location links, hence the *estimated* signal strength values used to construct the input graph edges. Moreover, most city-wide network scenarios specify a desired level of coverage for their target area, e.g., 95% outdoor coverage. Therefore, we require a probabilistic coverage constraint, where a fraction of client locations must obtain signal strength above the threshold θ_a. For this work, we assume the signal strength estimation accuracy and the coverage requirement is 95%. Note that due to the plurality of access tier links available at a coverage location, there is also a probability that an alternative mesh node is reachable at the desired signal strength θ_a. Our results indicate that with 95% of links accurately estimated, less than 3% of target locations are not covered, over 97% of target locations receive coverage. In this formulation, the signal strength metric measures the quality of a link, which restricts the scope from considering congestion and contention effects, and we assume channel assignment is handled separately to enhance spatial reuse.

D. Connectivity Constraint

The backhaul tier connects each mesh node to a gateway, directly or via multi-hop paths through other mesh nodes. When gateway locations are unknown or not yet selected, we account for any possible gateway configuration with the constraint that each mesh node must have a path to all other mesh nodes. This full connectivity ensures gateway reachability regardless of a gateway's location. Correspondingly, the *connectivity constraint* requires that the undirected graph derived from the vertices in P is *connected*, where an edge exists between two chosen mesh node locations if the estimated signal strength is greater than the threshold θ_b for backhaul links. In the second case where gateway locations are known, we require there to be a path from each target coverage location to at least one gateway. Section III describes how our algorithm ensures connectivity if gateway locations are known a priori.

E. Capacity Constraint

Wireless bandwidth is shared amongst all clients, and as a result, it is often desirable to limit the number of potential sharers of the scarce wireless spectrum. Our formulation enforces this by imposing a maximum degree b_v on the connectivity of a mesh node, where the vertex bound b_v is homogeneous for all mesh node locations. Intuitively, we are restricting the number of locations each mesh node serves, and the degree bound also limits the number of other mesh nodes that each mesh node is connected to. This constraint will be critical in the proof of a the bound on our algorithm's worst-case performance. More complete capacity formulations take into account heterogeneous user demands and interference [16], but we do not consider these scenarios in this paper.

III. Placement Algorithms

This section introduces the Degree-Constrained Terminal Steiner tree (DCTST) problem and presents our two proposed placement algorithms. The novel features of our algorithms are 1) a discrete-graph input formulation to incorporate non-uniform propagation, and 2) minimizing deployment size with a constant-factor approximation ratio proven using our new DCTST problem formulation.

A. Steiner Tree Framework

The Degree-Constrained Terminal Steiner tree problem is a special case of the Steiner tree problem in graphs [14]. The Steiner tree problem in graphs involves finding a minimum weight tree that spans the regular vertices in the input graph. In contrast to a simple spanning tree, though, there is an additional set of discrete vertices, termed *Steiner Points*, that are selectively added to further decrease the total weight of the solution spanning tree. We design our algorithms upon a framework where the regular vertices represent target coverage locations and the Steiner Points map to the potential mesh node installation locations. We build a modified Steiner tree, a *Degree-Constrained Terminal Steiner tree*, where all regular vertices (coverage locations) are required to be a leaf in the solution Steiner tree, mirroring the fact that client devices do not act as traffic relays in a mesh network. We also add an additional constraint on the maximum degree of any Steiner point, capturing the limit on the number of clients a mesh node can simultaneously serve. An example DCTST is shown in Figure 2.

The construction of a DCTST on the input connectivity graph solves the challenge of jointly providing target coverage and mesh connectivity as follows. Like prior work, connectivity is satisfied by requiring the chosen mesh locations to form a tree that spans all mesh nodes, connecting the backhaul tier. Unlike prior work, we satisfy the coverage constraint by requiring the spanning tree to also include all target client coverage locations as leafs in the tree. Additionally, the degree constraint enforces a capacity limitation, such that no single mesh node serves a disproportionately large area. A DCTST construction algorithm outputs the set of chosen Steiner Points, which we use to indicate node deployment locations that satisfy coverage, capacity, and connectivity constraints.

The Degree-Constrained Steiner tree problem in graphs has been shown to be NP-hard and recent research has developed polynomial time approximation algorithms for Steiner tree problems with degree bounds. The DCTST is also NP-hard because the terminal constraint enforces two different degree bounds: terminals have degree of exactly one and Steiner Points have degree at most b_v. The current best known degree-constrained Steiner tree algorithm [10] has constant factor approximation ratio of 2, and this work is the first to show results for the DCTST formulation, i.e. with both degree and terminal constraints.

DCTSTs provide a framework for mesh node placement, upon which we build our resulting placement algorithms. First, we present the algorithm **Minimize-Nodes**, which finds the minimum number of deployed mesh nodes. This algorithm operates on an input graph of estimated link signal strengths. Secondly, we build an enhanced algorithm, **Measure-and-Place**, that uses the first algorithm as an inner loop and iteratively refines the input connectivity graph to ensure that

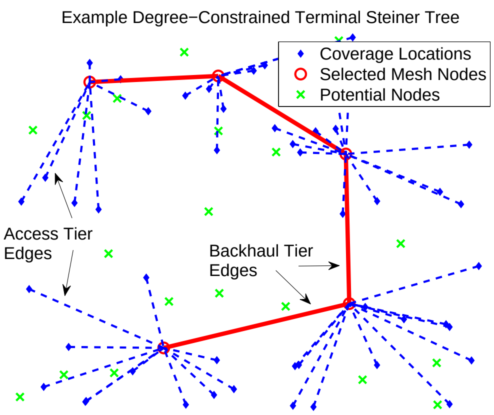

Fig. 2. Example Degree-Constrained Terminal Steiner tree. Mesh nodes (Steiner Points) form solid red backhaul graph and coverage locations connect to tree with dashed blue lines. Also shown are potential mesh node locations (shown as x's) that were not chosen by the DCTST algorithm.

all backhaul links are measured above the acceptable signal strength threshold.

B. Algorithm Minimize-Nodes

For the first algorithm, we minimize the number of deployed mesh nodes with constant-factor approximation ratio by solving the DCTST problem. Our approach contrasts with prior work on placement algorithms which achieve constant-factor approximation only in uniform propagation scenarios [4], whereas our bounds apply to the more general case of non-uniform propagation. The mesh node placement problem can also be formulated as a version of the *node-weighted* Steiner tree problem, but we do not take this approach because the best known node-weighted algorithms have logarithmic worst-case bounds [13]. Instead, we focus on the *edge-weighted* formulation and show that a constant-factor approximation can be achieved for the *node minimization* problem.

A set of gateway locations in the mesh network is often known a priori due to physical and practical constraints on the locations of wired connections. With known gateway positions, the mesh connectivity constraint requires that each target coverage location have a path to at least one gateway node (thereby connecting to the Internet). We augment the input connectivity graph in two ways. First, we add one additional node to the graph representing the wired Internet, with equally weighted edges to each gateway location. This new node is designated as a terminal node, requiring it be spanned by a Steiner tree. As a result, all valid Steiner trees must include the gateway node locations as they are the only nodes with edges to the Internet node. Second, we also add zero-weight edges between each gateway location, causing all gateways to be chosen as a mesh node by our algorithms. The *triangle inequality* is not violated as no triangles can be drawn with two zero-weight edges; observe that edges between a gateway and other mesh nodes are considered equivalent to an edge between two mesh nodes.

Table I presents the Minimize-Nodes algorithm pseudocode, which builds the appropriate input connectivity graph and then builds a DCTST in order to select the mesh node deployment locations. The time complexity of this polynomial-time algorithm is dominated by that of the DCTST algorithm, which itself is dominated by the underlying Steiner tree algorithm's complexity. The best-known algorithm [10] solves the problem using an iterative LP rounding technique. We next prove a constant-factor approximation ratio for the node minimization objective, where the constant factor is function of the node capacity (i.e. degree bound).

C. Constant Factor Approximation Ratio

In an instance of the DEGREE CONSTRAINED TERMINAL STEINER TREE (DCTST) problem, we are given a complete graph $G = (V, E)$ with metric $d : V \times V \rightarrow \mathbb{R}$ on the edges, a set of terminals $C \subseteq V$, degree bounds b_v for each vertex $v \in V \setminus C$ and the task is to a find a minimum weight tree which spans C such that the degree of each vertex $v \in V \setminus C$ is at most b_v and each vertex in C is a leaf. Let $deg_T(v)$ indicate the degree of vertex v in tree T and let $d(T)$ indicate the total weight of the edges in tree T.

We prove the following theorem regarding the DCTST problem.

Theorem 1 *There exists a polynomial time algorithm which given an instance of the* DEGREE CONSTRAINED TERMINAL STEINER TREE *problem returns a Terminal Steiner tree F of weight no more than $\frac{7}{2}$ times the weight of the optimal tree. Moreover, the degree $deg_F(v)$ of each vertex $v \in V \setminus C$ in tree F is at most $2b_v + 6$.*

Prior work by Lau and Singh presented algorithmic results for the degree constrained Steiner tree problem, i.e. with no terminal constraint. The following theorem is from [10], and we use this result to prove Theorem 1.

Theorem 2 *There is a polynomial time algorithm which given an instance of the degree constrained Steiner tree problem returns a Steiner tree T of weight no more than twice the optimal such that the degree bound of each vertex is violated by at most an additive constant* 3.

Proof of Theorem 1: Given an instance of the DCTST problem, we formulate an instance of the degree constrained Steiner tree problem where we set the degree bound of each vertex $v \in C$ to be 1 and for each vertex $v \notin C$ to be b_v. Moreover, we remove all edges (u, v) where $u, v \in C$. Observe that such an edge is not used by an optimal solution. Hence, T^*, the optimal solution to DCTST, remains a feasible solution to the instance of the degree constrained Steiner tree problem. Using Theorem 2, we obtain a tree T whose weight is at most $2d(T^*)$ such that $deg_T(v) \leq 4$ for each $v \in R$ and $deg_T(v) \leq b_v + 3$ for each $v \in V \setminus C$. Observe that T is not a terminal Steiner tree since some vertices in C need not be leaves. Let $v \in C$ be such a vertex, and let the neighbors of v in T be $u_1, \ldots, u_k$ where $k \geq 2$ and $u_i \in V \setminus C$ for each $1 \leq i \leq k$. Let (v, u_k) be the edge incident at v in T with largest weight. Replace the edges incident at v by edges $\{(v, u_1), (u_1, u_2), \ldots, (u_{k-1}, u_k)\}$. The weight of the new edges equals $d(v, u_1) + d(u_1, u_2) + \cdots + d(u_{k-1}, u_k)$.

Now, by the triangle inequality and the property of arithmetic averages, we have that

$$d(v,u_1)+d(u_1,u_2)+\cdots+d(u_{k-1},u_k) \leq \\ d(v,u_1)+d(v,u_1)+d(v,u_2)+\cdots+d(v,u_k) = \\ 2d(v,u_1)+2d(v,u_2)+\cdots+2d(v,u_{k-1})+d(v,u_k) \leq \\ \frac{2k-1}{k}[d(v,u_1)+\cdots+d(v,u_k)]$$

which is at most $\frac{7}{4}[d(v,u_1)+\cdots+d(v,u_k)]$ since $k \leq 4$. Applying this procedure at each terminal vertex, we obtain a terminal Steiner tree F of weight no more than $\frac{7}{4}w(T) \leq \frac{7}{2}w(T^*)$ as claimed. Moreover, each edge incident at a vertex $v \in V \setminus C$ is replaced by at most two edges incident at vertex v. Hence, the degree of v is at most $2(b_v+3) = 2b_v+6$ in F as claimed. □

Next, we show that the DCTST formulation can also be used to minimize the number of deployed mesh nodes. Recall that DCTST is an *edge*-weighted problem, whereas the **Minimize-Nodes** placement algorithm seeks to minimize the number of deployed *nodes*. There are two key points used to show that the constant-factor approximation holds for our algorithm: 1) the fact that each target coverage location is connected to exactly one mesh node in the tree T and 2) mesh nodes have a fixed capacity that can be expressed by a degree bound b_v.

We take advantage of the first fact by assigning all *usable* (estimated to be above threshold) access tier edges the same weight. In other words, the estimated signal strength values are *not* used as edge weights, but rather used to determine which edges are usable.

For a network with n target coverage locations, the coverage constraint requires each of the n locations to have exactly one link in the solution DCTST. We set all usable access tier edges to have uniform weights (i.e., normalized to one), resulting in a constant weight n in all valid solutions. Similarly, the total weight due to backhaul links is $(m-1)$ where m is the number of deployed mesh nodes in the final solution. Let n^* and m^* represent the values of n and m in the optimal solution. From Theorem 1, our DCTST algorithm's approximation ratio is 3.5, which we write as the bound on the ratio of our solution edge weight to the optimal edge weight:

$$\frac{m+n}{m^*+n^*} \leq 3.5$$

As per the previous observation that the number of access tier edges in all valid DCTSTs is identical, let $n = n^*$. Rearranging terms:

$$\frac{m}{m^*} \leq 3.5 + 2.5\frac{n}{m^*} \qquad (1)$$

The rightmost term in the above equation would grow linearly with the size of the input. To address this, we recall that the capacity constraint b_v enforces an upper bound on the degree of all the nodes in the Steiner tree: $(n/m) \leq b_v$. As a result, the rightmost term in Equation (1) is a constant upper bounded by the number of coverage locations supported per mesh node. By choosing edge weights where the backhaul edges have twice the weight of the access edges, we are able to preserve the triangle inequality and further halve the value of the constant in the above equation from 2.5 to 1.25. Theorem 3 now follows.

Theorem 3 *Minimize-Nodes is a polynomial time algorithm to find the minimum number of mesh nodes to deploy with approximation ratio of* $1.25b_v$ *and degree violation of at most* $2b_v+6$.

```
Create connectivity graph G from input SNR graph H, s.t.
  Backhaul tier edges exist if SNR estimate > θ_b
  Access tier edges exist if SNR estimate > θ_a
  If gateway locations known,
   augment G with shortcuts between gateways
  Set backhaul tier edge weight to 2
  Set access tier edge weight to 1
Run DCTST algorithm on G
Output chosen Steiner Points P
```

TABLE I
ALGORITHM MINIMIZE-NODES

While the above algorithm has a constant-factor approximation ratio for a given b_v, it is important to note that this constant may be large, depending on the capacity of mesh nodes. In other words, an increased mesh node capacity will result in a larger value for the constant in the approximation ratio. We note that this is the first constant-factor approximation for the placement problem with non-uniform propagation, and our practical-case evaluations in Section IV show that our algorithm outperforms prior techniques by up to 80%. Therefore, while choosing the capacity bound impacts the worst-case bound, our empirical results indicate the practical performance is high over a wide range of values of b_v.

D. Algorithm Measure-and-Place

The algorithm **Measure-and-Place** addresses the uncertainty in the estimation of the input link graph and the corresponding fact that all link signal strengths cannot be known without measurements. To do this, we enhance the Minimize-Nodes algorithm presented previously with additional interactive measurements in order to ensure all backhaul links in the solution DCTST are *measured* to be above the threshold. In other words, we avoid relying on estimated link signal strengths for the critical backhaul links of the deployed network.

There are two challenges in using interactive measurement feedback: how to keep small the number of links to measure and how to use the specific measurement data to inform the final placement decision. Note that we differentiate this feedback with any training measurements used in the initial signal strength estimation process. We address the problem of keeping measurement overhead low by measuring each *backhaul* link in the minimum weight DCTST chosen with the Minimize-Nodes algorithm. As a result, we only measure links that are estimated to be above threshold and picked as candidate backhaul links by our algorithm. At each iteration, the number of measurements taken does not exceed one less than the number of selected mesh nodes. Note that we focus on measuring backhaul links as they aggregate traffic from the access tier and are therefore more performance critical, though the same methodology extends to measuring selected access tier edges as well.

With measurement information obtained, we then address the challenge of how to utilize the measurement results to iteratively refine the input graph and achieve our objective. Our key technique is to not only remove poor links, but also to decrease the weight on above-threshold links, increasing the chances that these links will be chosen in the next iteration of our algorithm. More specifically, let M_l represent the measured signal strength on link l, let w represent the default backhaul edge weight, and let ϵ be a small, positive constant where $\epsilon \ll w$. Then, for each link measured, we modify the input graph edge weight in one of the following four ways:

1) If link l is unmeasured and estimated below θ_b, remove edge from graph (equivalent to weight $= \infty$).
2) If link l is unmeasured and estimated above θ_b, set edge weight to w.
3) If measurement $M_l \geq \theta_b$, set edge weight to $(w - \epsilon)$.
4) If measurement $M_l < \theta_b$, remove edge from graph.

Table II outlines the operation of the Measure-and-Place algorithm.

We set ϵ to a small, positive number in order to give preference in the next iteration to the links measured above the acceptable threshold. Also, by making ϵ small, we do not significantly change the magnitude of the weights in the resulting Steiner tree, but rather use the modified weights as a tie-breaking mechanism. Therefore, a small ϵ value does not impact the relative size of terms in Equation (1). This enhanced placement algorithm minimizes the number of deployed mesh nodes, subject to the measurement information available. The full version of this problem would be to jointly minimize the number of deployed nodes and measurements, but this formulation does not have any provable algorithmic bounds. The Measure-and-Place algorithm completes when all backhaul links in the solution DCTST have been measured and confirmed to be above threshold θ_b. Note that to ensure the algorithm finds a valid solution, the algorithm can lower the performance thresholds θ_a and θ_b when the only usable links were incorrectly estimated as low quality links.

Mark all backhaul edges in input graph G as unmeasured
Initial solution node set $P = \emptyset$, tree $T = \emptyset$

Do {
  Use Minimize-Nodes algorithm (see Table I),
    obtaining solution nodes P and spanning tree T
  Measure all un-measured backhaul edges in T
  Update edge weights in G
} while ($\exists e \in T$, s.t. e is backhaul and $M_e < \theta_b$)
Output solution P and T as measurement-validated

TABLE II
ALGORITHM MEASURE-AND-PLACE

IV. PLACEMENT EVALUATION

This section evaluates the performance of our proposed placement algorithms, **Minimize-Nodes** and **Measure-and-Place**. We compare the proposed placement algorithms with geometric covering algorithms in a non-uniform propagation setting based on measured propagation data from the currently deployed GoogleWiFi mesh network.

A. Evaluation Methodology

The input to the placement algorithms consists of a topology of potential mesh node locations and target coverage locations, signal strength estimations for each location pair, and signal strength conformance thresholds θ_a (access tier) and θ_b (backhaul tier). The physical-layer connectivity graph used in our evaluation matches the measured coverage regions shapes in the deployed GoogleWiFi network (see Figure 1 for examples).

We evaluate algorithms on a regular city-block topology, the GoogleWiFi network in Mountain View, CA [15]. Figure 3 shows the potential mesh node locations in part of the 7.25 km^2 urban neighborhood considered. In this topology, a potential mesh node location is at each street light post and target coverage locations are chosen uniformly with 5 meter spacing. The density of potential mesh node locations is approximately 200 locations per km^2. We also evaluated our algorithms with randomly generated topologies and the findings are similar.

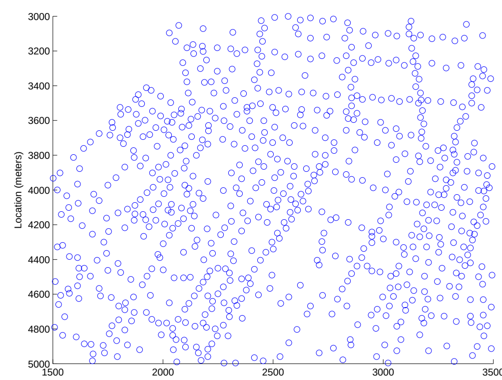

Fig. 3. GoogleWiFi neighborhood topology with circles indicating potential mesh node locations, i.e. lamp posts. Target coverage locations (not shown) are uniformly spaced through the region.

For evaluation purposes, the DCTST algorithm uses a heuristic version of the Steiner tree algorithm [14], which improves computational efficiency on the studied topologies. The placement algorithm we compare against is a two-phase geometric algorithm [11], [17]. Two-phase algorithms first satisfy the coverage constraint by solving the geometric disc covering problem, and then add nodes to ensure connectivity by building a graph Steiner tree (not a DCTST). Specifically, we implement the discrete disc covering algorithm used in [17]. We then use a basic Steiner tree algorithm [14] to satisfy the connectivity constraint by letting the mesh nodes chosen in the covering phase be the regular vertices and then choosing additional mesh nodes (Steiner Points) to build a connected backhaul tier.

The disc covering algorithm is not intended for non-uniform propagation settings, therefore in all evaluations, we choose the disc radius giving the best result where the network is covered with least number of mesh nodes. In the uniform propagation case, the input connectivity graph G derives signal strength values using only the general path loss exponent from the GoogleWiFi measurement data.

B. Non-uniform Propagation Setting

To evaluate our algorithms with realistic propagation values, we employ measurements from a coverage study [15] in the GoogleWiFi network in Mountain View, CA. These measurements consist of signal strength readings taken at 35000 locations from a car-based laptop with external antenna. Using the terrain information from an economic zoning map results in propagation prediction accuracy of approximately 90%, and we then use this to estimate the propagation at any given potential mesh node location in the city. The average path loss value observed is 3.7, the average shadowing value is 8dB, and the reference SNR is measured to be 60 dB at 10 meters distance. There is considerable variation in the path loss exponent on different paths, between a value of 2 (line-of-sight) and above 6 (very poor propagation).

To generate the physical-layer connectivity graphs in the studied topologies, each potential mesh node location is marked at locations approximately every 40 meters along streets, corresponding to street lamp locations. For each location, we then generate the estimated signal strength values using the modified path loss equations (see Equations 1 and 2 in [15]) to estimate signal strength as a function of distance and intervening terrain features. The true measured signal strengths cannot be estimated perfectly, and hence we add shadowing as a zero-mean Gaussian random variable for each link to represent the true propagation value. The amount of standard deviation then determines the likelihood that the estimate incorrectly indicates if the link is above or below threshold.

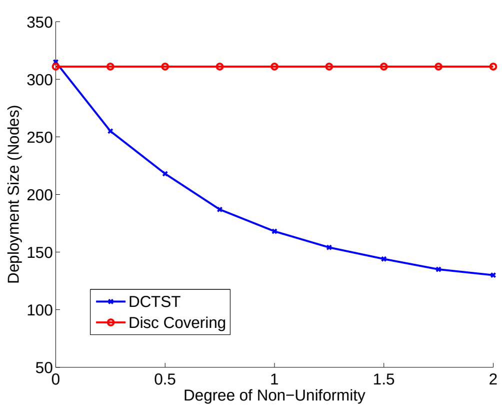

Fig. 4. Comparison of placement algorithm performance on GoogleWiFi topology with varying levels of non-uniform propagation. The x-axis is normalized to the measured propagation: 0 represents uniform propagation and 2 indicates that non-uniformity is increased by a factor of $2\times$.

We first evaluate how non-uniform propagation impacts our algorithms and the resulting network size. Figure 4 varies the non-uniformity of propagation and plots the deployment size resulting from Minimize-Nodes and a Disc Covering algorithm. The x-axis represents the degree of non-uniformity and is calculated by variably dampening or exaggerating the difference between each link's propagation and the median propagation in our data set. More specifically, for all links, we find the average path loss, which then allows us to compute for each link the difference in path loss from the average. We then weight this difference to lessen or enhance the level of non-uniformity. Observe that a weight of zero leads to uniform propagation and a weight of one leads to the actual measured propagation. The average (median) coverage range is kept constant and therefore the disc covering algorithm's result does not change.

With non-uniform propagation, the DCTST algorithm outperforms disc covering by up to 80%, whereas disc covering is slightly (less than 2%) better in the uniform propagation setting. This result is not surprising, as both algorithms were designed for different propagation settings, and even with small amounts of non-uniformity, our proposed algorithm outperforms disc covering. Note that there is an accuracy and cost tradeoff when increasing the disc radius in non-uniform settings. For the measured propagation setting (x-axis value of 1), a disc covering solution of the same size as our DCTST solution (168 nodes) results in approximately $4\times$ as many coverage holes, i.e. 95% coverage versus 80.6% coverage.

The more surprising result in Figure 4 is that the magnitude of the DCTST solution size decreases significantly as non-uniformity increases, indicating that non-uniform propagation in the GoogleWiFi topology is advantageous in network planning. The reason for this is first that mesh nodes provide coverage to larger areas in non-uniform propagation settings, largely due to the fact that area covered is a function of radius squared. Also, because propagation is based on terrain features, the DCTST algorithm takes advantages of well-placed node locations (e.g., no obstacles nearby). Secondly, the pattern of potential locations in the GoogleWiFi topology further increases the benefit of non-uniform propagation because the links between street lamps are strongly line-of-sight. As a result, in non-uniform settings, each mesh node has a larger degree of connectivity to its neighboring mesh node locations. We consider settings where only one of the access or backhaul tiers have uniform propagation and find that approximately 70% of the gain is from non-uniform propagation at the access tier.

We next to examine the impact on the availability of potential mesh node locations, using a subset of the GoogleWiFi topology. Figure 5 plots the resulting network size using the algorithm Minimize-Nodes and the disc covering algorithm. The performance difference between Minimize-Nodes and the disc covering algorithm increases as the density of available locations increases. This occurs because the disc covering algorithm is limited by the inaccuracy of using circles to approximate non-uniform coverage regions. Even with the larger networks from disc covering, the number of coverage holes is $3\times$ higher than in the network constructed by the algorithm Minimize-Nodes. When the number of potential mesh node locations reaches a factor of $5\times$ the number ultimately deployed, additional potential locations have negligible

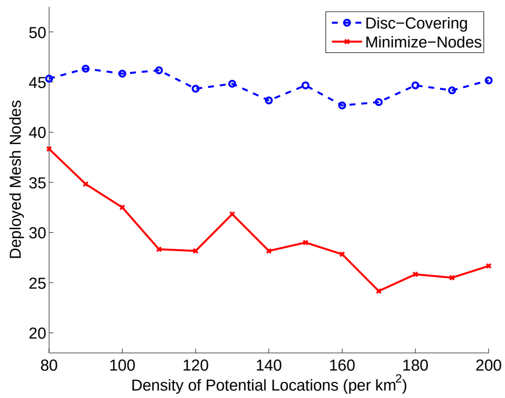

Fig. 5. Comparison of placement algorithm performance on GoogleWiFi topology as a function of the density of available mesh node locations.

benefit.

Figure 6 plots the resulting network size when varying the coverage footprint size, i.e., the threshold θ_a of the access tier, with three different values of the backhaul link transmission range. A smaller access tier range leads to the coverage becoming the limiting factor in deployment planning. Figure 6 also plots the placement size of a coverage-only network, i.e., without the requirement of backhaul connectivity. Note that the backhaul range of 158 meters is the default value used in the GoogleWiFi scenario and access tier ranges are most often smaller than backhaul. In these practical cases, our results indicate that the mesh network with a backhaul tier requires no more than 15% more nodes than a coverage-only network. This suggests that the mesh network is a more effective broadband wireless architecture than an outdoor WLAN (coverage only) due to the small number of additional nodes needed versus the cost of installing wired connections.

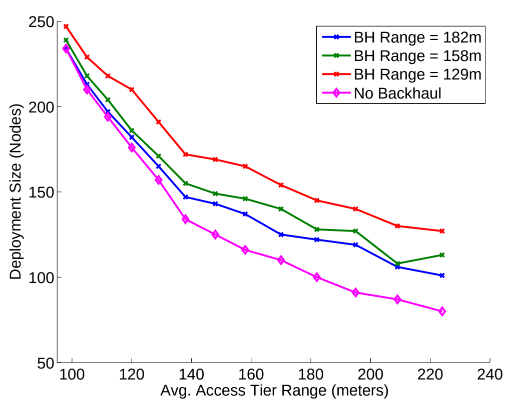

Fig. 6. Comparison of placement algorithm performance on GoogleWiFi topology with varying access tier ranges and backhaul (BH) ranges to show how network hardware influences placement.

We next consider the performance of the second proposed algorithm, Measure-and-Place, restricting our results to a smaller section of the GoogleWiFi topology. For comparison, we also use a greedy measurement algorithm to ensure backhaul connectivity, where each measured link with sufficient signal strength is always kept as part of the solution Steiner tree. We investigate the performance of these algorithms as a function of backhaul link shadowing, which is a zero-mean Gaussian random variable modeling variation of link qualities. Higher shadowing values correspond to less accurate signal strength estimation.

Figure 7 plots the number of measurements needed to ensure backhaul connectivity using the Measure-and-Place algorithm and the number of deployed mesh nodes resulting from the Measure-and-Place algorithm. As seen, the iterative measurement algorithm requires approximately $3\times$ more measurements than the number of deployed mesh nodes. The greedy measurement algorithm requires slightly fewer measurements because it always keeps a link measured to be above threshold. As a consequence, though, it requires a larger number of deployed nodes, as it does not adjust the DCTST at each iteration to balance the measurements and network size. With the Measure-and-Place algorithm, the number of measurements needed is of the same order as the number of deployed mesh nodes. For reference, an exhaustive measurement study of all links shorter than 500 meters requires $1000\times$ more measurements.

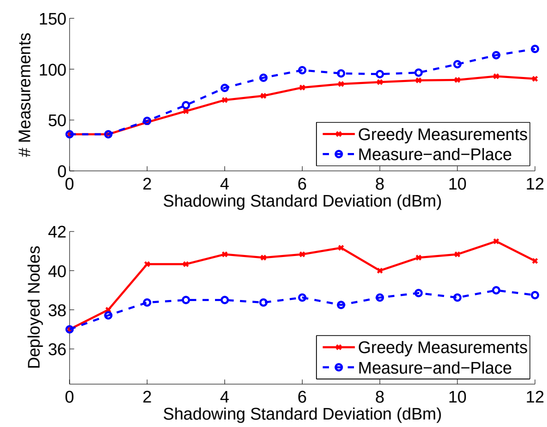

Fig. 7. The number of feedback measurements required to ensure backhaul tier connectivity using algorithm Measure-and-Place in the GoogleWiFi topology.

V. Related Work

Prior work on placement algorithms for wireless mesh networks has focused on heuristic algorithms. Integer programming [2] and greedy heuristics [5] have been proposed to choose locations for mesh nodes, but these approaches have no provable bounds on worst case performance and do not consider non-uniform propagation. Similarly, the problem of backbone construction for multi-hop ad hoc networks also requires solving coverage and connectivity constraints. In the context of these networks, a two-phase, disc covering approximation algorithm [17] has been presented, but only for the uniform propagation setting.

There has been much work on the closely related problem of placing relay nodes in a two-tier sensor network scenario [20]. Previous papers have taken a geometric approach to solving the coverage problem using a disk-covering algorithm, and then separately ensuring backhaul connectivity [11]. Most recently, a two-phase polynomial time approximation scheme has been proposed for relay node placement in uniform propagation scenarios [4]. In contrast, our proposed algorithms jointly solve coverage and connectivity and apply to the more general non-uniform propagation case. Other approximation algorithms for relay placement in the uniform propagation setting have been proposed to satisfy connectivity with constrained node placement [12] or to provide redundancy through the placement of two nodes at each deployment location [21].

Wireless LAN and cellular networks present a related base station placement problem, though proposed approximation algorithms [18] do not require the connectivity constraint seen in mesh networks. For placement in WLAN scenarios, heuristic algorithms [7] and integer programming techniques [8] have been proposed for non-uniform propagation scenarios, but these algorithms do not provide worst-case bounds and do not address the problem of reducing the number of measurements needed. Similarly, algorithms for cellular base station placement for various objectives have been presented using heuristic algorithms or integer programs [19].

Graph Steiner tree problems have been studied extensively and are closely related to the MNP problem. The graph Steiner tree problem involves finding a minimum weight spanning tree over the set of terminals and a chosen subset of Steiner Points. Prior two-phase placement algorithms have used Steiner tree algorithms to solve the connectivity constraint, after using the geometric facility location problem [1] for coverage. The Degree-Constrained Steiner tree problem adds an upper bound on the degree of any node in the Steiner tree, and has been solved using LP rounding techniques that result in an approximation ratio of 2 and a degree constraint violation of no more than 3 [10]. The node weighted version of the degree constrained Steiner tree has a best-known algorithm with logarithmic approximation ratio [13]. The Terminal Steiner tree problem is a special-case of the Steiner tree problem where all the terminal nodes are required to be leafs in the solution spanning tree. The best known algorithm for the Steiner tree problem has approximation ratio of 1.55 [14], and the edge-weighted Terminal Steiner tree problem has approximation ratio of 3.1 [3]. These algorithms, though, do not allow a constant-factor approximation ratio for the mesh node placement problem because they do not have a degree constraint.

VI. Conclusions

This paper presents a new graph-theoretic formulation and approximation algorithms for the mesh node placement problem. We first formulate the placement problem for non-uniform propagation scenarios as a graph-theoretic problem, in contrast to prior geometric disc covering formulations. The key advantage of this formulation is that it allows for per-link signal strength specification using either realistic physical-layer propagation models or measurement results. We then present mesh placement algorithms to *jointly* solve for coverage, capacity, and connectivity constraints, through the construction of a Degree-Constrained Terminal Steiner tree on an input connectivity graph consisting of both coverage locations and potential mesh node locations. As a result, our algorithm is the first-known constant factor approximation ratio algorithm for the problem of minimizing deployment size. The Minimize-Nodes algorithm minimizes deployed nodes using the estimated signal strength values in the input graph and then the Measure-and-Place algorithm iteratively measures a small number of backhaul links in the solution DCTST. As a result, our algorithm ensures that the backhaul-tier is fully connected in the final deployment without requiring an exhaustive measurement study. We then evaluate the performance of our algorithms, showing an 80% improvement in measured non-uniform propagation settings.

References

[1] H. Alt, E. M. Arkin, H. Brönnimann, J. Erickson, S. P. Fekete, C. Knauer, J. Lenchner, J. S. B. Mitchell, and K. Whittlesey. Minimum-cost coverage of point sets by disks. In *Proceedings of the Symposium on Computational Geometry*, pages 449–458, 2006.

[2] E. Amaldi, A. Capone, M. Cesana, and F. Malucelli. Optimization models for the radio planning of wireless mesh networks. In *IFIP Networking*, May 2007.

[3] D. Drake and S. Hougardy. On approximation algorithms for the terminal Steiner tree problem. *Information Processing Letters*, 89(1), 2004.

[4] A. Efrat, S. P. Fekete, P. R. Gaddehosur, J. S. B. Mitchell, V. Polishchuk, and J. Suomela. Improved approximation algorithms for relay placement. In *European Symposium on Algorithms*, pages 356–367, September 2008.

[5] A. A. Franklin and C. S. R. Murthy. Node placement algorithm for deployment of two-tier wireless mesh networks. In *Proceedings of IEEE GLOBECOM*, Washington, D.C., November 2007.

[6] M. Iskander and Z. Yun. Propagation prediction models for wireless communication systems. *IEEE Transactions on Microwave Theory and Techniques*, 50(3):662–673, Mar 2002.

[7] K. Jaffrès-Runser, J.-M. Gorce, and S. Ubéda. Mono- and multiobjective formulations for the indoor wireless LAN planning problem. *Computers and Operations Research*, 35(12):3885–3901, 2008.

[8] M. Kamenetsky and M. Unbehaun. Coverage planning for outdoor wireless LAN systems. In *International Zurich Seminar on Broadband Communications*, Zurich, Switz., February 2002.

[9] R. Karrer, A. Sabharwal, and E. Knightly. Enabling large-scale wireless broadband: the case for TAPs. In *Proceedings of HotNets-II*, Cambridge, MA, Nov. 2003.

[10] L. C. Lau and M. Singh. Additive approximation for bounded degree survivable network design. In *Proceedings of ACM STOC*, Vancouver, Canada, May 2008.

[11] E. L. Lloyd and G. Xue. Relay node placement in wireless sensor networks. *IEEE Transactions on Computing*, 56(1), 2007.

[12] 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 *Proceedings of IEEE INFOCOM*, Phoenix, AZ, April 2008.

[13] S. S. Ravi, M. V. Marathe, D. J. Rosenkrantz, S. S. Ravi, and H. B. Hunt. Approximation algorithms for degree-constrained minimum-cost network-design problems. *Algorithmica*, 31(1):58–78, 2001.

[14] G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19(1), 2005.

[15] J. Robinson, R. Swaminathan, and E. Knightly. Assessment of urban-scale wireless networks with a small number of measurements. In *Proceedings of ACM MobiCom*, San Francisco, CA, September 2008.

[16] J. Robinson, M. Uysal, R. Swaminathan, and E. Knightly. Adding capacity points to a wireless mesh network using local search. In *Proceedings of IEEE INFOCOM*, Phoenix, AZ, April 2008.

[17] A. Srinivas, G. Zussman, and E. Modiano. Mobile backbone networks – construction and maintenance. In *Proceedings of ACM MobiHoc*, 2006.

[18] K. Tutschku. Demand-based radio network planning of cellular mobile communication systems. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, March 1998.

[19] R. Whitaker and S. Hurley. Evolution of planning for wireless communication systems. *IEEE Hawaii International Conference on System Sciences*, 09, 2003.

[20] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655, 2008.

[21] W. Zhang, G. Xue, and S. Misra. Fault-tolerant relay node placement in wireless sensor network: Problems and algorithms. In *Proceedings of IEEE INFOCOM*, Anchorage, AK, May 2007.