RICE UNIVERSITY

Deployment and Assessment of Wireless Mesh Networks

by

Joshua Robinson

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE DOCTOR OF PHILOSOPHY

APPROVED, THESIS COMMITTEE:

Edward W. Knightly, Chair
Professor of Electrical and Computer Engineering

David B. Johnson
Professor of Computer Science and Electrical and Computer Engineering

Ashutosh Sabharwal
Assistant Professor of Electrical and Computer Engineering

Ram Swaminathan
Senior Research Scientist at HP Labs

Houston, Texas
March, 2009
ABSTRACT

Deployment and Assessment of Wireless Mesh Networks

by

Joshua Robinson

Multi-tier wireless mesh network deployments are a popular, cost-effective means to provide wireless broadband connectivity to neighborhoods and cities. Client devices within the coverage area of a mesh network connect wirelessly to fixed mesh nodes, which then forward traffic directly or via multi-hop paths to capacity injection points. The small number of capacity points act as Internet gateways and reduce overall network cost by limiting the amount of costly wired infrastructure needed. Non-uniform wireless signal propagation and the contention caused by multi-hop traffic contribute the challenge of deploying mesh networks with both high performance and low cost. This dissertation presents and evaluates cost-efficient algorithms for deployment planning and measurement-based assessment of wireless mesh networks.

The mesh node placement problem requires mesh nodes to provide ubiquitous network coverage to clients, as well as connectivity amongst mesh nodes. The first contribution of this thesis is to present a graph-theoretic formulation of the NP-hard
mesh node placement problem. This is the first formulation which considers the case in outdoor networks where signal propagation is non-uniform and enables the design of graph-theoretic approximation algorithms in order to minimize the deployment size or average contention. Secondly, deployment planning must select locations for the placement of capacity points, as their locations determine the path lengths in the networks and the resulting capacity available to transmit data to and from the Internet. To choose capacity point locations, I first present a technique to efficiently calculate network capacity and then two local search algorithms adapted from solutions to the facility location problem. Third, this thesis presents a framework for the measurement-based verification of a deployed network’s performance. To avoid relying on expensive and exhaustive measurement studies, I consider the assessment problem with a limited number of measurements. The framework uses terrain-informed estimation, per-node virtual sectorization, and measurement refinement to accurately predict the network’s performance at any given location. I evaluate the presented algorithms on realistic network topologies and with a large-scale measurement study of two currently deployed mesh networks: the TFA network and GoogleWiFi network. The thesis results demonstrate the essential nature of incorporating measurements, realistic propagation, and wireless contention into mesh network planning and assessment techniques.
Acknowledgments

Graduate school has been a long journey and I owe many different people to have reached this far. I will try my best here to succinctly thank all who have helped and encouraged me.

First and foremost, I would like to thank my advisor, Dr. Edward Knightly, for the guidance, support, and opportunities he has provided me. He has been a model researcher and a great example to learn from, pushing me to excel in my research and studies while also allowing me to choose my own focus and path.

Ram Swaminathan has been an invaluable source of advice and knowledge as a co-advisor and mentor. Through our collaboration and my internship at HP Labs, Ram has taught me about optimization, algorithms, and life. I am confident my studies and career would not be as far as they are without his help. My collaboration with HP Labs has resulted in a majority of my thesis work, and I am also indebted to my co-author Mustafa Uysal for his advice and experience, as well as Rich Friedrich for making the internship possible.

I would also like to thank my thesis committee members. Ashu Sabharwal and Dave Johnson have both provided valuable questions and insights throughout my graduate studies, often including feedback for my work from a new perspective. I always left a presentation with new ideas. Behnaam Aazhang and Eugene Ng have
also helped on my Master's committee, all together providing new perspectives and ideas on my research.

The current and former members of the Rice Networks Group have helped me through conversations and collaborations, especially Joseph Camp. Joe has been a collaborator, colleague, and friend throughout graduate school and I am glad to have had his support over the last six years. I am also indebted to my other co-authors, Chris Steger and Dina Papagiannaki, for what I have learned from them about their respective areas of expertise. I always enjoyed weekly group meetings with the Rice Networks Group, learning from each of my fellow graduate students and postdocs: Ehsan, Tassos, Ahmed, Eugenio, Misko, Jingpu, Violeta, Soups, Aleksandar, Bahar, Vikram, Marco, Alessandro, Vincenzo, Omer, Theodoris, Michele, Roger, and of course, Dee. As much of my research has been focused on and inspired by the Technology-for-All network, I would also like to thank Will Reed, Jim Forrest, and Nell Warnes, who have championed and managed the TFA network. And finally, I began my research career as undergraduate, working with Dr. Knightly and Steve Sheafor, who both gave me an idea of research, and consequently a desire and reason to apply to graduate school.

My parents have always encouraged and supported my endeavors, and I am immensely appreciative of the platform they have given me to take my career and life wherever I want. I thank them most of all.
Contents

Abstract ii
Acknowledgments iv
List of Figures x
List of Tables xv

1 Introduction 1
1.1 Summary of Thesis Contributions 3
1.2 Thesis Overview 4

2 Background 7
2.1 Multi-tier Mesh Networks 7
2.2 Physical Layer for Mesh Links 9
2.3 Case-Study Mesh Networks 11
   2.3.1 TFA Network 11
   2.3.2 GoogleWiFi 12

3 Mesh Node Placement 15
3.1 Introduction 16
3.2 Placement Formulation 19
   3.2.1 MNP Connectivity Graph 19
3.2.2 Input: Non-Uniform Propagation ........................................ 20
3.2.3 Coverage Constraint .................................................. 21
3.2.4 Connectivity Constraint ............................................... 23
3.3 Placement Algorithms .................................................. 24
  3.3.1 Terminal Steiner Tree Framework .................................. 24
  3.3.2 Minimizing Number of Nodes ..................................... 27
  3.3.3 Algorithm Measure-and-Place .................................... 29
  3.3.4 Algorithm Minimize-Contention ................................ 31
  3.3.5 Algorithm Extensions ............................................. 33
3.4 Placement Evaluation .................................................. 38
  3.4.1 Evaluation Methodology .......................................... 38
  3.4.2 Non-uniform Propagation Scenario .............................. 40
  3.4.3 Measure-and-Place Evaluation ................................... 44
3.5 Related Work .......................................................... 46
3.6 Summary ............................................................... 49

4 Capacity Point Deployment .............................................. 50
  4.1 Introduction .......................................................... 51
  4.2 Adding Capacity Points ............................................. 55
    4.2.1 Gateway-Limited Fair Capacity ............................... 55
    4.2.2 Gateway Placement Problem .................................. 58
4.3 Solving the Placement Problem ........................................... 60
  4.3.1 Solving by Maximizing Capacity ................................... 60
  4.3.2 Solving by Minimizing Contention ............................... 65
4.4 Evaluation ...................................................................... 70
  4.4.1 Validating Capacity Calculation ................................. 70
  4.4.2 Performance of Placement Algorithms ....................... 73
  4.4.3 TFA Placements Case Study ......................................... 79
  4.4.4 Jointly Placing Mesh Nodes and Gateways ................. 83
4.5 Related Work ................................................................ 86
4.6 Summary ...................................................................... 88

5 Measurement Assessment of Deployed Networks .................. 89
  5.1 Introduction ................................................................... 90
  5.2 Assessment Framework .................................................. 93
    5.2.1 Problem Definition ................................................ 93
    5.2.2 Metric Definitions ............................................... 95
    5.2.3 Metric Sector Framework ..................................... 96
  5.3 Estimation and Refinement ............................................. 99
    5.3.1 Performance Metric Estimation ............................... 99
    5.3.2 Estimating Monotonic Metric Regions .................... 103
    5.3.3 Estimated Boundary Refinement ............................. 105
5.3.4 Non-Monotonic Metrics ........................................ 107

5.4 Framework Validation ............................................ 109
  5.4.1 Framework Validation Methodology .......................... 109
  5.4.2 Measurement Study Background .............................. 112
  5.4.3 Coverage Metric .............................................. 113
  5.4.4 Modulation Rate Metric ...................................... 122
  5.4.5 Coverage Redundancy Metric ................................ 124

5.5 Deployment Evaluation ........................................... 126
  5.5.1 Coverage Holes and Deployment Efficiency .................. 126
  5.5.2 Load-Balanced Node Deployment ............................. 129

5.6 Related Work ................................................... 133

5.7 Summary .......................................................... 136

6 Conclusions ....................................................... 137

6.1 Thesis Contributions ............................................ 137

6.2 Future Work ..................................................... 139

References .......................................................... 141
List of Figures

2.1 A multi-tier mesh network consists of three tiers. From right to left, the access tier connects client devices to infrastructure mesh nodes, the backhaul tier then interconnects mesh nodes to a capacity point, which then aggregates data to the Internet via a capacity link (e.g., fiber optic or WiMax). .................................................. 8

2.2 Topology of the TFA network including 20 mesh nodes, of which four are capacity points. Black lines represent dedicated directional wireless links as the capacity tier connections. ................................. 11

2.3 Topology of the GoogleWiFi network including approximately 450 mesh nodes, of which 60 are capacity points [1]. ................................. 14

3.1 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. .............................. 22

3.2 Example 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 that were not chosen by the TST algorithm. ................................. 26
3.3 TFA neighborhood topology with circles indicating households and businesses, which are both clients and potential mesh node locations. The centrally-located X's indicate target coverage locations at a neighborhood school. .......................................... 39

3.4 Comparison of placement algorithm performance on a random topology with non-uniform propagation as measured in the GoogleWiFi network. 42

3.5 Comparison of placement algorithm performance on random topology with varying access tier thresholds (ranges) to show how network hardware influences placement. ............................................. 43

3.6 The resulting network size and number of feedback measurements required to ensure backhaul tier connectivity using algorithm Measure-and-Place in the TFA topology. ............................................. 45

4.1 Validation of model with ns-2 simulation, placing one gateway on 4x4 grid topology and ranking each of the possible 16 configurations. . . 71

4.2 Capacity of placements arrived at by algorithms on a square grid topology with 49 mesh nodes. ................................................................. 74

4.3 Performance of placement algorithms on Chaska (top) and Google (bottom) topologies. ................................................................. 77

4.4 Comparison of MinHopCount and MinContention algorithms with optimal placement for TFA-projected topology. .................. 78
4.5 Histogram of all placements of 4 new gateways in the projected TFA topology, along with the capacities found by the evaluated algorithms. 80

4.6 Percentage of the total number of possible placements that result in a better capacity than the placement found by the proposed algorithms. 81

4.7 Scatter plot for all possible gateway placements in the TFA-proj topology with 4 new gateways. 82

4.8 Comparison of two-stage placement and joint placement of mesh nodes and gateways in the TFA neighborhood topology. 84

5.1 Example metric sectorization and boundaries for an example Google-WiFi mesh node's coverage region. 98

5.2 Terrain map of Mountain View, CA, for the measurement study area. 101

5.3 Example of sectorization process in algorithm Estimate-Mono-Metric-Region for a GoogleWiFi mesh node. As a function of angle from the mesh node, the plot shows the estimated ranges, the sectorized estimate, and the measured ranges. 105

5.4 Coverage map in the GoogleWiFi network with circles representing covered locations and x's representing holes in the 12 km² measured area with 168 mesh nodes. 113
5.5 Coverage region of the TFA network with circles representing covered locations and x's representing holes in 3 km² measured area, including the network edge where weak signal was measured. ................................................. 114

5.6 Predictive accuracy of network coverage assessment with measurement budget, comparing the assessment framework with grid sampling strategy. .............................................................................................................. 115

5.7 Predictive accuracy for the TFA and GoogleWiFi networks using uniform-propagation estimation, terrain-informed estimation, and ideal estimation. ........................................................................................................... 117

5.8 Violations of the monotonicity property. The probability that signal strength increases by more than 3 dB when increasing radius in a sector of width 0.1 degrees. ............................................................. 119

5.9 Accuracy of boundary refinement algorithm for the coverage region of mesh nodes in GoogleWiFi. ................................................................. 121

5.10 Comparison of boundary refinement strategies for modulation rate metric in GoogleWiFi, either directly measuring modulation rate values or estimating from coverage measurements. ......................... 123

5.11 Distribution of the number of covering mesh nodes (redundancy) for each location in the GoogleWiFi network, comparing measured values with estimations. ................................................................. 125
5.12 Probability of a coverage hole as a function of local deployment density for both networks. ........................................... 128

5.13 Percentage gain in average client throughput from using a centralized, load-balancing association scheme versus realistic, local association with strongest SNR. ........................................... 131
List of Tables

3.1  Algorithm Minimize-Nodes .................................................. 29
3.2  Algorithm Measure-and-Place .............................................. 32
3.3  Summary of proposed placement algorithms ............................. 37

4.1  Pseudocode for MinHopCount algorithm. .............................. 68
4.2  Pseudocode for MinContention algorithm. .............................. 69
Chapter 1
Introduction

The demand for wireless broadband Internet access has increased steadily due to both the ubiquitous nature of mobile devices and the low cost of commodity wireless equipment. Multi-tier wireless mesh networks provide Internet connectivity through the deployment of 802.11-based mesh nodes in multi-hop topologies [2]. With a mesh network, network operators lower infrastructure costs by eliminating expensive wires and instead relying on multi-hop wireless communication to forward traffic from clients to a small number of connected capacity points [3].

Mesh nodes are deployed such that they provide service to client devices via the access tier. In the backhaul tier, mesh nodes interconnect to forward client traffic to and from designated capacity points either directly or through multihop paths. In the capacity tier, each capacity point connects to the wired Internet using a separated communications link, such as directional antennas, WiMax, or fiber optics.

This thesis considers the problem of deploying a wireless mesh network and validating the achieved performance of a deployment. A network operator is able to choose the mesh topology with the goal of maximizing the performance of the resulting network. Choosing mesh network deployment locations is challenging because of the exponential number of possible configurations, i.e., it is impractical for a network
operator to measure all possible topologies before deployment. Therefore, network designers attempt to select optimal configurations based on performance models which capture the expected coverage and throughput capacity. Choosing the optimal location of the mesh nodes and capacity points is a generalization of NP-hard facility location problems, and so it is important to design polynomial-time approximation algorithms to find near-optimal solutions in feasible time.

A major factor that makes determining the performance of a mesh network difficult is non-uniform physical-layer propagation in outdoor environments. Therefore, it is crucial to understand the propagation of a deployment in order to verify coverage goals or identify poor-performance areas. Measuring all possible links after deployment is not feasible due to high cost and time requirements involved. Exhaustive measurements can be taken at all target coverage locations, but this technique does not scale to large networks or incremental network upgrades. Instead, this thesis focuses on the problem of predicting the deployed network's spatial performance using a limited number of measurements.

The objective of this thesis is to design and evaluate algorithms for deployment planning and measurement assessment of wireless mesh networks. The proposed solutions together form a toolkit for low cost mesh network deployment and measurement validation. In particular, I present algorithms for: 1) the deployment of mesh nodes, 2) the deployment of capacity points, and 3) the accurate assessment of spatial per-
formance with a limited number of measurements.

1.1 Summary of Thesis Contributions

The first topic of this thesis is the selection of ideal mesh node installation locations. The problem of selecting mesh node locations to jointly ensure client coverage and mesh connectivity is unsolved because of non-uniform physical-layer propagation, practical constraints on mesh node locations, and the impracticality of conducting full measurement studies. Prior algorithms for the related problem of relay placement in sensor networks use a two-stage approach and assume uniform propagation [4]. To place mesh nodes, I present graph-theoretic algorithms that incorporate both non-uniform propagation estimations and measurement results in order to minimize the number of deployed mesh nodes. Further, I present an algorithm extension to verify backhaul links with a minimal number of measurements before deployment.

The second aspect of this thesis focuses on placement of the critical capacity points, as their location and quantity determines the maximum throughput supported by the network. Namely, the placement of these points determines the hop-length of the paths in the network, the amount of congestion, and the available bandwidth to and from the Internet. Prior work on greedy heuristics [5] and local search operations [6] have been developed for gateway placement, but neither incorporates wireless contention nor studies incremental deployment. This thesis presents a computationally efficient technique to calculating network capacity and two local search algorithms
for choosing capacity point locations which incorporate wireless contention effects.

Lastly, a network operator must assess the achieved spatial performance of the network in order validate objectives or identify poor performance locations. The wireless network assessment problem characterizes the network’s metric regions, i.e., identifying locations in the network where the given performance metric meets a conformance threshold. Existing assessment strategies either require exhaustive measurements [7] or use detailed physical-layer object descriptions to precisely estimate propagation characteristics [8]. Unfortunately, these approaches are expensive and usually impractical, especially for incremental network upgrades. In this thesis, I present a framework for accurate estimation of metric regions using terrain-informed estimation and measurement refinement.

1.2 Thesis Overview

The remainder of this thesis is organized as follows. In Chapter 2, I describe the multi-tier mesh network architecture, wireless propagation basics, and introduce two examples of currently deployed mesh networks.

Chapter 3 presents my formulation for the mesh node placement problem, which is based on a general connectivity graph and accurately inputs non-uniform propagation. The thesis then presents approximation algorithms to minimize the number of deployed mesh nodes, which build upon algorithms for the Terminal Steiner tree problem and therefore jointly satisfies both coverage and connectivity constraints.
Also, this chapter discusses several extensions to the proposed algorithms, e.g., to minimize network contention or to iteratively measure backhaul links to verify connectivity. Lastly, I evaluate the algorithms' performance on realistic topologies based on the TFA neighborhood.

Chapter 4 first presents a computationally efficient technique for calculating the gateway-limited fair capacity, based on the observation that the capacity points will be the bottleneck point in a mesh network. I then adapt two local search algorithms for the facility location problem to solve the gateway placement problem, one of which minimizes wireless contention with constant factor approximation ratio. This chapters then evaluates the proposed algorithms and evaluates a case-study scenario in the TFA network to demonstrate the similarity of near-optimal solutions to the optimal solution, thereby showing the appropriateness of local search techniques.

In Chapter 5, I present a framework for accurate prediction of the spatial performance of a deployed wireless network with a limited number of measurements. The first contribution is to formulate this network assessment problem as estimating the shape of the network’s metric regions, i.e., the set of locations with performance meeting a given threshold. To accurately estimate metric regions, this thesis presents a framework incorporating terrain-informed estimation, per-node virtual sectorization, and measurement refinement. Utilizing large-scale measurement studies of two currently deployed networks, I then show that this technique achieves high accuracy
in real scenarios and identify non-monotonicity as the major source of prediction errors. Lastly, the thesis presents applications of an assessment study, studying the size and locations of coverage holes and load-balancing properties in the two mesh deployments. Finally, Chapter 6 concludes and discusses future research directions.
Chapter 2
Background

This chapter introduces the multi-tier mesh network architecture for providing wireless broadband access to large contiguous areas and then provides background on physical-layer signal propagation models.

2.1 Multi-tier Mesh Networks

Multi-tier mesh networks (Fig. 2.1) consist of three separate tiers. First, client devices connect directly to mesh nodes via the access tier. The mesh nodes are typically dedicated infrastructure nodes mounted in fixed positions. A subset of these infrastructure mesh nodes feature a connection to the Internet and are referred to as capacity points. Mesh nodes forward traffic to a capacity point via multi-hop paths on the backhaul tier. The third tier is the capacity tier, where capacity points aggregate traffic to the Internet via various dedicated links, such as directional wireless, WiMax, or fiber optics. The capacity tier is distinct from the backhaul tier as it does not contend with backhaul tier traffic.

In contrast, the wireless nodes in an ad hoc wireless network are the end users of the network, whereas this thesis considers infrastructure networks, which have the objective of providing service to end-user client devices. Note that this definition of an infrastructure mesh network does not include community wireless networks
Figure 2.1  A multi-tier mesh network consists of three tiers. From right to left, the access tier connects client devices to infrastructure mesh nodes, the backhaul tier then interconnects mesh nodes to a capacity point, which then aggregates data to the Internet via a capacity link (e.g., fiber optic or WiMax).

such as Roofnet [9] because they do not include an access tier. A single-tier wireless networks consists of only the access tier, with all nodes directly connected to the wired Internet, and is equivalent to a wireless LAN (WLAN) architecture. Note that mesh nodes feature omni-directional antennas in most deployment scenarios, though directional or sectorized antennas are also possible. Further, the distinction between two-tier and multi-tier networks is the potential use of dedicated, contention-free wireless connections to help aggregate backhaul traffic on the capacity tier.
2.2 Physical Layer for Mesh Links

An significant challenge in planning mesh networks is that physical-layer link performance is irregular and often not predictable as it depends on complex interactions with obstacles in the environment. The performance and capacity of a mesh network depends heavily on the quality of the underlying wireless links, therefore it is necessary to accurately model propagation behavior.

Pathloss describes the attenuation experienced by a wireless signal at a given distance [7]. It has been found empirically that received signal power decays exponentially with distance. The rate of signal power decay, or pathloss exponent, depends heavily on the characteristics of the environment, such as location and composition of buildings and trees.

The pathloss exponent defines the general trend of signal strength decay for links in similar environments, but there is also variation in signal strength on links of the same distance. Shadowing describes these small variations in pathloss due to the randomness of most environments. In other words, shadowing takes into account the difference in signal strength on two links of equal distance. Prior empirical measurements have found shadowing to behave as a zero-mean Gaussian random variable added to the received signal power. Shadowing standard deviation $\sigma_e$ is smallest in a homogeneous target environment. By incorporating shadowing, all estimates of received signal strength are inherently probabilistic, and it is this inability to ex-
actly predict signal strength levels that motivates the incorporation of non-uniform propagation models.

The received signal strength at a given location depends on the distance $d$ to the transmitter in question. Incorporating both pathloss and shadowing, the received signal strength $P_{dBm}$ is given as a function of distance $d$, a reference power level at distance $d_0$, the pathloss exponent $\alpha$, the shadowing term $\epsilon$, and the shadowing standard deviation, $\sigma_\epsilon$.

$$P_{dBm}(d) = P_{dBm}(d_0) - 10\alpha \log_{10}(\frac{d}{d_0}) + \epsilon$$  \hspace{1cm} (2.1)

$$\epsilon \sim N(0, \sigma_\epsilon^2)$$  \hspace{1cm} (2.2)

When signal strength threshold values are used to distinguish usable links, the threshold value is 25 dB, a value that experimentally corresponds to 2 Mbps throughput \textit{(not datarate)} with 802.11b cards. This rate corresponds to the peak target rate offered to end users in the case-study mesh deployment.

Note that Equation (2.1) calculates the expected signal strength of a link, not the instantaneous signal strength at any specific point in time. In reality, the link also experiences fast fading \cite{7} and varies with time, resulting in changes in the throughput capacity or loss rate of the link. For this thesis, the measurements and estimations of pathloss are assumed to be time-averaged values.
2.3 Case-Study Mesh Networks

Two different operational mesh networks serve as basis for measurements and topologies in this thesis. This section describes the architectures and hardware of the Technology-for-All (TFA) and GoogleWiFi networks.

2.3.1 TFA Network

The TFA Network is an urban mesh network, deployed in southeast Houston by Rice University [10] for free Internet access. The network is in a 4.2 km² residential neighborhood with heavy tree coverage. At the time of the measurement study [11], the network consisted of 17 mesh nodes, providing coverage to a 3 km² area. Figure 2.2 displays the current TFA topology map.

![Figure 2.2](image)

*Figure 2.2* Topology of the TFA network including 20 mesh nodes, of which four are capacity points. Black lines represent dedicated directional wireless links as the capacity tier connections.

The mesh nodes feature a mini-PC with VIA C3 x86-based process running at 1
GHz and a custom Linux operating system. The access and backhaul tiers operate with a single 802.11b wireless card with 200 mW transmission power and a 15 dBi external omnidirectional antenna. The antennas are mounted at 10 meter height, which is higher than most of the houses and some of the trees in the neighborhood. There are three capacity points in the network, which are connected back to the wired gateway via 802.11g directional links. The TFA measurement platform was a laptop inside a car with an 802.11b wireless interface, a 7 dBi external antenna, and a GPS receiver.

2.3.2 GoogleWiFi

GoogleWiFi is a multi-tier mesh network operated by Google in Mountain View, CA. At the time of this study, the GoogleWiFi network consists of 447 Tropos mesh nodes mounted mostly on city light posts and covering a total outdoor area of 31 km², as shown in Figure 2.3. The access and backhaul tiers consist of Tropos mesh nodes, each with a 7.4 dBi antenna and a single 802.11g wireless interface. Of the GoogleWiFi mesh nodes, 58 are capacity points, connected to the Internet via long-range, line-of-sight directional antennas and non-802.11 radio devices. The client measurement platform for the GoogleWiFi study was a laptop with an external 802.11g wireless adapter, 3 dBi antenna, and GPS receiver.

The two networks have several key structural differences. The antennas used in TFA are taller and have higher gain, indicating a larger coverage region. This is offset
partially by the difference in terrain, as the TFA network is filled with larger, denser trees which act as attenuators. Moreover, the GoogleWiFi nodes are mounted on light poles along streets whereas most TFA nodes are mounted against houses in the interior of a residential block.

There are other mesh networks, such as the MadMesh network in Madison, WI, that have been studied [12]. A commonly referenced example network is the Roofnet network [9], though it does not meet this definition of a mesh network because it does not feature an access tier.
Figure 2.3  Topology of the GoogleWiFi network including approximately 450 mesh nodes, of which 60 are capacity points [1].
Chapter 3
Mesh Node Placement
3.1 Introduction

Wireless mesh networks provide Internet access to large contiguous areas through the placement of mesh nodes [3]. 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. In this chapter, I present graph-theoretic approximation algorithms to place nodes in the non-uniform propagation scenario in order to 1) minimize the number of deployed mesh nodes with small number of measurements or 2) minimize average contention per client and provide a constant factor bound on the worst case solution.

The first contribution of this chapter 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 formulation to consider the non-uniform propagation scenario by specifying connectivity based on per-link estimated signal quality, as opposed to prior work [13, 14, 4] which considers the idealized uniform propagation scenario. In other words, I 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, I use physical-layer estimation techniques [15, 16] to specify the potential links in the input connectivity graph. Propagation estimation, though, introduces estimation errors in the input connectivity graph, and a second key advantage of my formulation is to allow measurement-driven refinement of the input connectivity graph on a per-link basis, eliminating possible estimation errors.

I then show that the mesh node placement problem is NP-hard and consequently design polynomial-time approximation algorithms to choose placements, i.e., algorithms with provable bounds on worst-case performance. First, I present a framework to jointly satisfy both client coverage and mesh connectivity constraints by coupling the discrete graph input with modified solutions to the *Terminal Steiner tree* (TST) problem [17, 18]. The key technique is to build a solution Steiner tree which spans all selected mesh nodes (connectivity) and has the additional constraint that all target client locations are connected as leafs of the tree (coverage), thereby jointly satisfying both constraints. The first approximation algorithm, **Minimize-Contention**, minimizes the average contention region size at client locations with a constant-factor approximation ratio by adapting TST algorithms and assigning edge weights which obey the triangle inequality. Thus, while prior algorithms also have constant-factor approximation ratios, the results in this thesis represent the first that apply in the non-uniform propagation scenario. The second algorithm, **Measure-and-Place**, minimizes the number of deployed nodes while also using a small number of measure-
ments to ensure that all selected backhaul links are connected. The key technique of this algorithm is iterative Steiner tree construction combined by refinement of the input connectivity graph via measurements of selected backhaul links.

Finally, I 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 physical-layer scenarios, I use signal strength measurements from 35,000 locations and 150 mesh nodes in the GoogleWiFi and TFA networks in order to evaluate the impact of estimation errors. The presented algorithms result in 80% fewer deployed mesh nodes and 3 times fewer coverage holes in the resulting deployments. Further, I 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 chapter proceeds as follows. Section 3.2 defines the mesh network model. Section 3.3 presents new placement algorithms and Section 3.4 then presents the evaluation of the proposed placement algorithms. Finally, Section 3.5 describes related work and Section 3.6 summarizes.
3.2 Placement Formulation

The objective of the MNP 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-based specification of potential physical-layer links, used as input to the placement problem. I then describe the two general constraints of coverage and connectivity, which define a functional wireless mesh network and must be satisfied by a valid mesh node placement.

3.2.1 MNP Connectivity Graph

This thesis formulates the input to the mesh node placement (MNP) problem as a connectivity graph with nodes corresponding to discrete locations and edges between locations that indicate the existence of usable links. This formulation is the first to consider non-uniform propagation scenarios 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 (pathloss exponent) for all nodes. More formally, I 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. I 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 include only regions the operator seeks to provide service.

The second aspect of the input vertices is the set of potential mesh node locations, \( \mathbf{M} \), 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 installing. The vertex set of the input connectivity graph is defined as \( \mathbf{V} = \mathbf{C} \cup \mathbf{M} \), the union of potential mesh node locations and coverage locations.

### 3.2.2 Input: Non-Uniform Propagation

The input connectivity graph \( \mathbf{G} \) is a complete graph, where the set of links, \( \mathbf{E} \), are assigned values based on the estimated physical-layer signal strength between all nodes in graph \( \mathbf{G} \). In other words, an edge is assigned a value equal to the expected value of the corresponding link’s signal strength. Edges are usable if the signal strength is above a signal strength threshold \( \theta_a \) for access tier links or \( \theta_b \) 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 3.1 plots nine examples of measured coverage regions, illustrating the degree of non-uniform propagation encountered in practice. The ex-
act 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 [15, 16]. 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 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 formulation presented here.

### 3.2.3 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 $\theta_a$. More formally, let $P \subset M$ be the set of locations selected for mesh node placement. I then require for all locations $c \in C$ that there exists one edge (link) between location $c$ and one of the mesh nodes in $P$ with estimated or measured signal strength greater than $\theta_a$.

A challenge in formulating the coverage constraint is that it is usually impractical
Figure 3.1  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.

to measure all possible mesh node and coverage location links, hence the estimated signal strength values in the input graph. Moreover, since most city-wide network scenarios specify a desired level of coverage, e.g., 95% outdoor coverage, it is not necessary to have perfect signal strength information. Therefore, I use the estimated signal strength values to satisfy the coverage constraint, resulting in a fraction of uncovered locations in a valid solution deployment. I show later that the number of uncovered locations is kept less than 3% due to the plurality of access tier links available at a coverage location. In other words, there is strong probability that at least one mesh node is reachable at the desired signal strength $\theta_a$. In this formulation,
the signal strength metric measures the quality of a link, which restricts the scope from considering congestion and contention effects, and I assume channel assignment is handled separately to enhance spatial reuse.

3.2.4 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, I account for any possible gateway configuration with the constraint that each mesh node can connect 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 the edges exist between two chosen mesh node locations if the estimated signal strength is greater than the threshold $\theta_b$. In the second case where gateways are known, I relax the backhaul connectivity and instead require a Steiner forest, each tree including a gateway. The proposed algorithms, presented in the next section, focus on the more conservative full connectivity constraint.
3.3 Placement Algorithms

The proposed algorithmic framework for the MNP problem couples the discrete-graph input formulation with algorithmic results for the Terminal Steiner tree (TST) problem. This section first introduces the main component of the placement algorithm framework: Steiner trees and approximation algorithms for the TST problem. Using this framework of TSTs, I present an approximation algorithm to minimize the number of deployed mesh nodes and through measurement-driven feedback, ensure that all backhaul links exceed the acceptable signal strength threshold $\theta_b$. Lastly, I discuss several different constraints and objectives that fit within the TST framework.

3.3.1 Terminal Steiner Tree Framework

The Terminal Steiner tree problem is a special case of the Steiner tree problem in graphs [18]. 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. The presented algorithms build upon a framework where the regular vertices represent target coverage locations and the Steiner Points map to the potential mesh node installation locations. Specifically, I build a modified Steiner tree, a 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. An example TST is shown in Figure 3.2.

The construction of a TST on the discrete input graph solves the challenge of jointly providing coverage and 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, I satisfy the coverage constraint by requiring the spanning tree to also include all target client coverage locations as leafs in the tree. A TST construction algorithm outputs the set of chosen Steiner Points, which I use to indicate node deployment locations that satisfy coverage and connectivity constraints. Prior work satisfies the connectivity constraint by the construction of a Steiner Tree, whereas a contribution of this work is the observation that coverage can also be satisfied through a TST, as opposed to geometric disc covering algorithms. The connectivity graph is static in the TST problem, and I first assume that the signal strength estimation graph is deterministic and later relax this assumption to incorporate estimations and measurement feedback.

The TST problem in graphs has been shown to be NP-hard and recent research has developed polynomial time approximation algorithms for Steiner tree problems (TST algorithms employ Steiner tree algorithms as a building block). Specifically, there exists constant factor approximation algorithms which find a minimum weight TST within a provable constant factor of the optimal solution. The current best known TST algorithm has constant factor approximation ratio equal to 3.1 [17, 18].
Figure 3.2  Example 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 that were not chosen by the TST algorithm.

TSTs provide a framework for mesh node placement and a building block for the presented placement algorithms. First, I 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, I 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 all backhaul links are measured above the acceptable threshold. And third, I present an algorithm to minimize contention using edge weights which obey the triangle inequality.
3.3.2 Minimizing Number of Nodes

I first describe the basic placement algorithm, Minimize-Nodes, which finds a valid placement with the minimum number of deployed mesh nodes. Variants of the Steiner tree problem which minimize the number of Steiner points do not permit constant-factor approximation algorithms. Instead, I present an approximation algorithm for choosing node locations based on minimizing the edge weight of a TST. I show that, through selection of edge weights, the algorithm minimizes the number of deployed nodes with provable worst-case performance. Table 3.3.2 presents the Minimize-Nodes algorithm. The time complexity of this polynomial-time algorithm is dominated by the TST algorithm, which is itself dominated by the underlying Steiner tree algorithm's complexity.

The Minimize-Nodes algorithms minimizes the number of deployed mesh nodes within provable bound of the optimal, as described next. The main idea behind the technique is the fact that each target coverage location is connected to exactly one mesh node and therefore the total number of access tier links in the final solution is constant. I take advantage of this 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. Let all backhaul tier edge weights be \( u \) and all access tier edge weights be \( v \). 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 TST, resulting in a constant factor weight of $vn$ in all valid solutions. Similarly, the total weight due to backhaul links is $u(m - 1)$ 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. Based on the current best TST algorithm’s approximation ratio of 3.1, I first write the approximation ratio as the bound on the ratio of the actual solution to the optimal solution:

$$\frac{um + vn}{um^* + vn^*} \leq 3.1$$

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

$$\frac{m}{m^*} \leq 3.1 + 2.1 \frac{vn}{um^*} \quad (3.1)$$

In order to obtain a constant-factor approximation ratio, the chosen weights must satisfy $u \gg v$ so that the right-most term in Equation (3.1) goes to zero, but this would violate the necessary triangle inequality. Instead, I choose weights $u = 2$ and $v = 1$, which gives the desirable property of preserving the triangle inequality. Therefore, the algorithm minimizes the number of deployed mesh nodes with approximation ratio proportional to $\frac{n}{m^*}$. 
Create weighted graph $L$ from input graph $G$, s.t.
Backhaul tier edges exist if estimate $> \theta_b$
Access tier edges exist if estimate $> \theta_a$
Set backhaul-tier edge weight to $u = 2$
Set access-tier edge weight to $v = 1$
Run TST algorithm on $L$
Output chosen Steiner Points $P$

**Table 3.1** Algorithm Minimize-Nodes

### 3.3.3 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, I enhance the Minimize-Nodes algorithm presented previously with additional interactive measurements in order to ensure all backhaul links in the solution TST are *measured* to be above the threshold. In other words, this extension avoids 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 I differentiate this feedback with the training measurements used in the initial signal strength estimation process. I address the problem of keeping measurement overhead low by measuring each *backhaul* link in the minimum weight TST chosen with the Minimize-Nodes algorithm. As a result,
I only measure links that are estimated to be above threshold. Additionally, each iteration requires a number of measurements equal to no more than the number of selected mesh nodes minus one. Note that I 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, I then address the challenge of how to utilize the measurement results to iteratively refine the input graph and achieve the objective. My 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 the algorithm. Let $M_l$ represent the measured signal strength on link $l$ and let $u$ represent the backhaul edge weight. Then, for each link, I modify the input graph edge weight in one of the following four ways:

1. If link $l$ is unmeasured and estimated below $\theta_b$, remove edge from graph (equivalent to an infinite weight).

2. If link $l$ is unmeasured and estimated above $\theta_b$, set edge weight to $u$.

3. If measurement $M_l \geq \theta_b$, set edge weight to $(u - \epsilon)$.

4. If measurement $M_l < \theta_b$, remove edge from graph.

Table 3.3.3 outlines the operation of the Measure-and-Place algorithm.
The parameter $\epsilon$ is a small, positive number (much less than $u$), which gives preference in the next iteration to the links measured above the acceptable threshold. Also, by making $\epsilon$ small, it does not significantly change the magnitude of the weight in the resulting Steiner tree, but rather uses the modified weights as a tie-breaking mechanism. Therefore, a small $\epsilon$ value also does not impact the relative size of terms in Equation (3.1).

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 permit provable worst-case bounds. The Measure-and-Place algorithm completes when all backhaul links in the solution TST have been measured and confirmed to be above threshold $\theta_b$. Note that to ensure the algorithm finds a valid solution, the algorithm can lower the performance thresholds $\theta$ when the only usable links were incorrectly estimated as low quality links.

3.3.4 Algorithm Minimize-Contention

In a mesh network with a single radio hardware configuration, I present an algorithm that places nodes in order to minimize the average contention at each coverage location in the network. In other words, the contention-minimization algorithm evenly deploys mesh nodes so that coverage overlaps as little as possible and more simultaneous transmissions are possible. To achieve this objective, this algorithm uses
Mark all backhaul edges in input graph as unmeasured
Initial solution $P = \emptyset$, tree $T = \emptyset$

While $P$ non-empty and
$\exists e \in T$, s.t. $e$ is backhaul and unmeasured
{
    Use Minimize-Nodes algorithm (see Table 3.3.2)
    Update solution nodes $P$ and spanning tree $T$
    Measure all un-measured backhaul edges in $T$
}
Output solution $P$ and $T$ as measurement-validated

**Table 3.2** Algorithm Measure-and-Place

contention values as edge weights which I then show obey the triangle inequality.

I define contention with respect to a link as the number of node locations that
cannot receive packets while the bi-directional link in question is active. I use the
input connectivity graph $G$ to derive the contention graph for the access and backhaul
tiers, using the signal strength estimates and a contention threshold. Note that only
contending coverage locations count as the final mesh node locations are not known
a priori and so cannot be used to calculate edge weights before the algorithm is run.
The challenge in assigning contention values as edge weights is that the weights must
obey the triangle inequality in order for TST approximation ratio results to hold.
Therefore, I assign a link weight for each link in the input connectivity graph as
the total number of coverage locations that contend with either endpoint of the link,
which preserves the triangle inequality for contention and is shown next.
Consider a triangle of three node locations, $a$, $b$, and $c$, and the resulting three links between them, labeled $AB$, $AC$, and $BC$. The contention caused by link $AB$ is less than the sum of the contention of links $AC$ and $BC$ for the following reason. Let function $\Gamma()$ represent the number of nodes in contention with a node or link. By definition, link $AB$'s contention consists of the nodes in contention range of nodes $a$ and $b$, resulting in $\Gamma(AB) = \Gamma(a) \cup \Gamma(b)$. The contention caused by links $AC$ and $BC$ is $(\Gamma(a) \cup \Gamma(c)) + (\Gamma(b) \cup \Gamma(c))$ and is smallest when node $c$ contends with no mesh nodes. Therefore the contention is lower-bounded by $\Gamma(a) + \Gamma(b)$, which is greater than or equal to $\Gamma(AB)$, ensuring that the triangle inequality is preserved.

This contention minimization algorithm does not result in the network with the smallest number of mesh nodes if the case arises where adding an extra mesh node reduces the total contention. This case occurs if the input graph contains heterogeneous node types, with some featuring a much larger communication range (e.g., higher antenna mast). These long-range nodes would cover more target locations but also introduce significantly more contention, and the above algorithm seeks the minimal-contention deployment with these heterogeneous choices.

### 3.3.5 Algorithm Extensions

The TST-based algorithmic framework incorporates several different objectives and constraints, overviewed next. I introduce the following algorithm versions that: 1) maximize signal strength 2) provide redundancy, 3) incorporate a capacity constraint,
and 4) relax connectivity requirements. Adding these constraints builds on techniques for iterative refinement of the input connectivity graph $G$ in order to modify the output of the algorithms. Note that these additional constraints and the associated iterative refinement augment the Minimize-Nodes and Measure-and-Place algorithms. Table 3.3.5 summarizes the main features of the presented algorithms and extensions.

**Maximizing the Minimum Signal Strength**

The next objective presented is to maximize the quality of the backhaul links subject to a cost constraint. As this constraint does not obey the triangle inequality in graphs, I use iterative techniques to refine the input connectivity graph as follows. The challenge in this objective is to refine the input graph so that I use the highest-quality set of backhaul links. My approach is to iteratively increase the acceptable signal strength thresholds $\theta_a$ and $\theta_b$ until the given budget constraint is violated. This algorithm runs in polynomial time because I restrict the range of valid threshold values and only consider whole number threshold values. I use this assumption because most commercially available wireless interfaces report only a discrete set of signal strength values.

**k-Redundancy Constraint**

I next add the constraint of providing $k$-redundancy so that each coverage location is covered by $k$ mesh nodes. Note that I do not consider redundancy satisfied by two
mesh nodes in the same physical location (as in [19]) because node failures are often due to local conditions, such as power failure or physical damage.

My algorithmic approach is again to iteratively refine the input graph in order to build a redundant placement. The insight for this algorithm is to refine the input graph to iteratively build TSTs that together provide the desired redundancy. The algorithm works by removing edges already chosen and forcing the algorithm to choose new edges and mesh nodes, which leads to a redundant placement by a union of the nodes selected at each iteration.

The first iteration proceeds with the chosen objective, e.g., minimizing deployment size, producing an initial placement $P_1$. I then identify all coverage locations with $k$-redundancy from placement $P_1$ and remove the corresponding vertices from the input graph for the remaining iterations. For the remaining coverage locations, I remove each edge which connects to a mesh node in set $P_1$, marking the remaining locations with the degree of redundancy so far achieved. I use the modified input graph to find a new minimum weight TST and resulting placement $P_i$, for $i = 2 \ldots k$, repeating until all coverage locations are $k$-redundant. The final placement is the union of the selected mesh nodes at each iteration of the algorithm, i.e. $P = \bigcup_{i=1}^{k} P_i$.

The redundancy algorithm has no provable worst case performance bounds as the TST-based algorithm does not directly solve the redundant deployment problem. To ensure that this algorithm always find a solution if a valid one exists, I only remove
already served target coverage locations and the corresponding access-tier edges at each iteration. This ensures that all backhaul nodes and backhaul links can be chosen at each iteration. In other words, none of the input graph modifications preclude the choice of a set of mesh nodes and backhaul links.

**Capacity and Connectivity Constraints**

The capacitated version of the MNP problem corresponds closest to the connected facility location problem which has no known constant-factor approximation algorithms due to the fact that there is no metric service cost function for this problem. I address this challenge by adding a new constraint that requires a bounded degree on all nodes in the solution TST. This constraint limits the number of mesh nodes and coverage locations connected to any one mesh node and thus effectively controls the area served per mesh node. The algorithm greedily adds new mesh nodes until the node degree bounds are met.

As discussed in Section 3.2, the connectivity constraint requires a fully connected graph. This is a restrictive constraint in the sense that it allows for all possible gateway locations. The TST based algorithms can handle less restrictive formulations when some gateways are pre-placed by changing the algorithm that converts a Steiner tree to a TST. This conversion [17] takes the output of the Steiner Tree algorithm and greedily adds Steiner Points to result in a fully connected backhaul graph. If gateway locations are known, this conversion instead needs only to add Steiner Points to result
in a forest of connected trees each rooted at each gateway.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Technique Used</th>
<th>Approx. Ratio</th>
<th>Previous Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimize-Nodes</td>
<td>Assign weights $\ell$</td>
<td>$\frac{n}{m}$</td>
<td>Only for unif. propagation case</td>
</tr>
<tr>
<td></td>
<td>$u$ and $v$</td>
<td>$(n \text{ clients})$</td>
<td></td>
</tr>
<tr>
<td>Measure-and-Place</td>
<td>Iterative</td>
<td>None</td>
<td>Not considered</td>
</tr>
<tr>
<td>Min-Contention</td>
<td>Edge weights 3.1</td>
<td></td>
<td>Not considered</td>
</tr>
<tr>
<td>k-Redundancy</td>
<td>Iterative</td>
<td>None</td>
<td>Not considered</td>
</tr>
</tbody>
</table>

**Table 3.3** Summary of proposed placement algorithms
3.4 Placement Evaluation

This section evaluates the performance of the proposed placement algorithms, Minimize-Nodes and Measure-and-Place. I compare the proposed placement algorithms with geometric covering algorithms in non-uniform propagation scenarios based on measured propagation data from two currently deployed mesh networks.

3.4.1 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 node pair, and signal strength conformance thresholds $\theta_a$ and $\theta_b$. The physical-layer connectivity graph used in the study matches the real, measured coverage regions shapes in the deployed GoogleWiFi network (see Figure 3.1).

I evaluate algorithms on both random topologies and regular city-block topologies. The random topologies represent the case where the available locations for mesh node deployment are randomly distributed throughout a neighborhood or city area. For these random topologies, I generate the spacing of potential mesh node locations using a spatial Poisson process with intensity equal to the desired density of potential mesh node locations.

For the regular city-block topology, I consider the topology of the TFA mesh network in Houston, TX [10]. The TFA topology consists of a 4 km² urban neighborhood, part of which is shown in Figure 3.3. In this topology, a potential mesh node
location is at every house or business in the neighborhood, plus there are additional locations at schools and an apartment complex which are only target coverage locations. The density of potential mesh node locations is approximately 900 locations per km², where the spacing between houses is 15 to 20 meters. In this case, each house which acts as both a target coverage location and a potential location for mesh node installation.

![Figure 3.3](image)  
**Figure 3.3**  TFA neighborhood topology with circles indicating households and businesses, which are both clients and potential mesh node locations. The centrally-located X's indicate target coverage locations at a neighborhood school.

For comparison, I use a two-phase geometric algorithm [13, 14]. 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 TST). Specifically, I implement the discrete disc covering algorithm used in [14]. The Steiner tree algorithm [18] satisfies 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 scenarios, therefore in all evaluations, I choose the disc radius giving the best result where the network is covered with least number of mesh nodes. In the uniform propagation scenario, input connectivity graph $G$ derives signal strength values using only the general pathloss exponents. The specific values of these radii are chosen using measurement data from prior GoogleWiFi and TFA studies.

3.4.2 Non-uniform Propagation Scenario

To evaluate our algorithms with realistic propagation values, I employ measurements from a coverage study [16] 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. I characterize the observed 100 mesh nodes with the most thorough measurements and use their coverage footprints as templates in the experiments. Each mesh node’s coverage region is split into 15 degree sectors and the measurements within each sector are used to obtain a pathloss exponent and shadowing standard deviation value for each sector. The average pathloss value observed is 3.7, the average shadowing value is 8dBm, and the reference power is
measured to be $-32\text{dBm}$ at 10 meters distance. There is considerable variation in
the pathloss exponent of each sector, between a value of 2 (line-of-sight) and nearly
6 (very poor propagation).

To generate the physical-layer connectivity graphs in the studied topologies, each
potential mesh node location is randomly assigned to one of the GoogleWiFi coverage
templates. For each link, I then generate the estimated signal strength using the
pathloss exponent and standard pathloss equations [7] to estimate signal strength as
a function of distance and angle (i.e., the sector's pathloss). The true measured signal
strengths cannot be perfectly estimated, and hence I add shadowing as a zero-mean
Gaussian random variable for each link. The amount of standard deviation then
determines the likelihood that the estimate incorrectly indicates if the link is above
or below threshold.

Figure 3.4 plots the resulting network size using the algorithm Minimize-Nodes
and the disc covering algorithm. Compared to the uniform propagation scenario, the
advantage of the algorithm is significantly increased, leading to solution deployments
that are 80% smaller than with the disc covering algorithm. Also, note that Minimize-
Nodes produces improving solutions as the density of available locations increases as
faster rate than the disc covering algorithm. The reason for this is that 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
Figure 3.4  Comparison of placement algorithm performance on a random topology with non-uniform propagation as measured in the GoogleWiFi network.

The number of coverage holes is 3x higher than in the network constructed by algorithm Minimize-Nodes.

Figure 3.5 plots the resulting network size when varying the coverage region size, i.e., the threshold $\theta_a$ of the access tier, with the backhaul link threshold held constant at $\theta_b = -80$ dBm. A higher threshold value (less negative) indicates the access range is shorter, and the result is that coverage becomes the limiting factor in deployment planning. As a result, the algorithm's performance gain increases to as high as 75% in the case of relatively small access tier ranges. The two-phase algorithms perform closer to the proposed algorithm in the opposite case because a relatively large access...
tier range leads to a situation where a covered network is automatically connected. Figure 3.5 also plots the placement size of a coverage-only network, i.e., just the first phase of the two-phase algorithm. This result reinforces the previous point, as in the relatively small access range case, the coverage-only deployment is the same size as the two-phase solution. In other words, connectivity was provided with no extra mesh nodes needed in the algorithm's second phase.

![Graph](image.png)

**Figure 3.5** Comparison of placement algorithm performance on random topology with varying access tier thresholds (ranges) to show how network hardware influences placement.

There are two sources of improvement in the Minimize-Nodes algorithm: jointly solving both constraints and accounting for non-uniform propagation. To determine the relative impact of both factors, I next consider the uniform propagation scenario,
isolating the gain due to jointly satisfying coverage and connectivity in one algorithm. The backhaul-tier range is set at 370 meters and the access tier range is 310 meters, using the same relative ranges found previously in TFA [10]. The results in the TFA network topology indicate that Minimize-Nodes results in 20% fewer deployed mesh nodes, i.e., the gain due to a joint algorithm. For the topology and propagation characteristics, directly accounting for non-uniform propagation has the greater impact on deployment planning.

3.4.3 Measure-and-Place Evaluation

I next consider the performance of the second proposed algorithm, Measure-and-Place, using the TFA topology and non-uniform propagation. As comparison, I also use a greedy measurement algorithm to ensure backhaul connectivity. This greedy algorithm functions by also measuring each link in the backhaul TST, except that the greedy algorithm always keeps in the solution a link that is confirmed to have signal strength above the threshold. I 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 3.6 plots the number of measurements needed to ensure backhaul connectivity using the Measure-and-Place algorithm. To complete an exhaustive measurement study and measure all potential links shorter than 500 meters requires 1000× more
measurements. Figure 3.6 also shows 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 TST 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.

![Figure 3.6](image)

**Figure 3.6** The resulting network size and number of feedback measurements required to ensure backhaul tier connectivity using algorithm Measure-and-Place in the TFA topology.
3.5 Related Work

Placement algorithms for wireless mesh networks have previously received interest. Integer programming [20] and greedy heuristics [21] 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. 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 approximation algorithm for the connected disc covering has been proposed [14], but this result only applies in the uniform propagation scenario. A further heuristic algorithm [22] has been presented with the objective of optimizing throughput, but also assuming uniform propagation, unrestricted node placement, and no connectivity constraint.

There has been much work on the closely related problem of placing relay nodes in a two-tier sensor network scenario [23]. Previous papers have taken a geometric approach to solving the coverage problem using a disk-covering algorithm, and then separately ensuring backhaul connectivity [13]. This prior algorithm has an approximation ratio of \((8 + \epsilon)\), where \(\epsilon\) is a small positive constant indicating the level of computational complexity. Additionally, this approximation ratio holds only in the case where the backhaul communication range is at least twice the access-tier range. Most recently, a two-phase polynomial time approximation scheme has been proposed for relay node placement in uniform propagation scenarios [4].
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 have been proposed to satisfy connectivity with constrained node placement [24] or to provide redundancy through the placement of two nodes at each deployment location [19].

Wireless LAN and cellular networks present a related base station placement problem, though proposed approximation algorithms [25] do not require the connectivity constraint seen in mesh networks. For placement in WLAN scenarios, heuristic algorithms [26] and integer programming techniques [27] 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. Also, heuristics have been proposed for a special case in WiMAX deployments [28]. Similarly, algorithms for cellular base station placement for various objectives have been presented [29, 30] using heuristic algorithms or integer programs. A survey of cellular planning literature can be found in [31].

The graph Steiner tree problem and the related Connected Facility Location problem 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 [32] for coverage. The Terminal Steiner tree (TST) problem is a special-case of the Steiner tree problem where all the terminal nodes are required to be leaves in the solution spanning tree. The best known approximation algorithm for the Steiner tree problem has approximation ratio of 1.55 [18]. The best known approximation algorithm for the TST problem is 3.1 [17]. The Connected Facility Location problem combines the Facility Location problem with Steiner trees to connect the opened facilities, and has yielded randomized algorithms with constant-factor approximation ratios [33]. These guarantees do not hold, though, in the MNP problem because there is no natural service cost function that lies in a metric space.
3.6 Summary

This chapter presents a novel graph-theoretic formulation and approximation algorithms for the mesh node placement (MNP) problem. I first formulate the MNP 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. I then present mesh placement algorithms to jointly solve for coverage and connectivity constraints, through the construction of a Terminal Steiner tree on a discrete input graph consisting of both coverage locations and potential mesh node locations. Resultingly, I present an algorithm which has a constant-factor approximation ratio (3.1) for the problem of minimizing the average contention per coverage location. A second presented algorithm minimizes deployed nodes using first the estimated signal strength values in the input graph and then iteratively measuring a small number of backhaul links in the solution TST. As a result, this algorithm ensures that the backhaul-tier is fully connected in the final deployment without requiring an exhaustive measurement study. Finally, I evaluate the performance of the proposed algorithms, showing an 80% improvement in measurement-based non-uniform propagation scenarios.
Chapter 4
Capacity Point Deployment
4.1 Introduction

A mesh network’s wired connected and dedicated wireless links connect the wireless mesh users with the wired Internet and are critical capacity points as their location and quantity determines the maximum throughput supported by the network. Namely, the placement of these points determines the hop-length of the paths in the network, the amount of congestion, and the available bandwidth to and from the Internet. Prior work has shown how capacity scales asymptotically with the number of gateways and nodes [34], but does not consider how to choose gateway locations in specific topologies. Likewise, greedy heuristics [5] and local search operations [6] have been developed for gateway placement, but neither incorporates wireless contention nor studies incremental deployment.

In this chapter, I formulate the gateway placement problem and then present and evaluate two local search algorithms. The gateway placement problem is related to the facility location and $k$-median problems and is NP-hard. Consequently, this thesis develops local search approximation algorithms in order to 1) provide the ability to apply local changes when incrementally upgrading a network without recomputing the full placement, and 2) provide polynomial time approximation schemes.

First, I propose an efficient technique to incorporate the effects of wireless contention and calculate the gateway-limited fair capacity of a wireless mesh network. While previous work provides a computational framework for capacity [35], I focus
here on access networks without direct client-to-client communication, i.e., networks in which all traffic traverses the gateway. Thus, capacity points necessarily carry more traffic than other mesh nodes, and consequently I define gateway-limited capacity in terms of the contention experienced at each gateway. This calculation technique is suitable for local search where a large space of possible operations must be considered and, for optimization purposes, the capacity can be separated into two components: path lengths and contention.

I next present two local search-based gateway placement algorithms adapted from the facility location problem. A key challenge is that the contention at each gateway depends on the full routing matrix. Therefore, each gateway’s capacity depends on the locations of other gateways and cannot be known in advance of determining other gateway placements. To address this challenge, the algorithms feature two different approaches to estimating the unknown gateway capacities. The first algorithm, termed MinHopCount, adapts a local search algorithm for the capacitated facility location problem [36] and iteratively estimates the unknown wireless gateway capacities. The idea of local search is to carefully choose a set of gateways to close and open a set of new gateways subject to capacity and budget constraints in order to minimize the objective of interest, i.e. the average hop count. Lowering hop count generally (but not always) increases capacity and has the additional critical property of obeying the triangle inequality. The second placement algorithm, MinContention,
adapts from a solution to the uncapacitated \( k\)-median facility location problem and minimizes the average contention for all mesh nodes with provable approximation ratio of \( 3 + \epsilon \) [37], where the \( \epsilon \) parameter determines accuracy and runtime. To minimize contention, I assign link weights equal to the amount of contention caused by each link, considering contention on all nodes instead of only gateways. Further, the link weight is the union of the set of nodes in contention range of either end of the link, which preserves the triangle inequality for \textit{swap}()-based local search. This local search is similar to open/close except that a \textit{swap} must open an \textit{equal} number of gateways as it closes. The MinHopCount algorithm is more general and can handle gateways with non-uniform costs, whereas the MinContention algorithm has a built-in budget constraint, and therefore retains a provable constant-factor approximation ratio.

Lastly, I evaluate the performance of the proposed algorithms in realistic topologies. I first validate that the capacity calculation techniques correctly rank placements, as compared to ranking placements based on measurement data from an operational urban mesh network. To compare placement algorithms, I perform numerical simulations on the topologies of three currently deployed mesh networks: Technology For All (TFA), Chaska, and GoogleWiFi.* I find that the local search algorithms perform up to 64% better than a greedy algorithm and produce placements within

2% of the optimal placement found via exhaustive search. I also study the degree of similarity between gateway placements using a hop-distance metric that measures the amount of change needed to transform one gateway placement to the other. I find that the relative distance between the optimal solution and near-optimal solutions found by local search is small, which indicates the suitability of local operations.

The remainder of this chapter proceeds as follows. Section 4.2 introduces the mesh capacity calculation technique and formally defines the gateway placement problem. Section 4.3 presents the two local search algorithms and Section 4.4 describes the evaluation of the placement algorithms. Section 4.5 discusses related work, and then Section 4.6 summarizes.
4.2 Adding Capacity Points

This section first introduces a new technique for calculating the gateway-limited fair capacity of a mesh network. I then formulate the problem of upgrading the capacity of an existing mesh network. For ease of discussion, I refer to all capacity points as "gateways" whether they are a true wireline gateway or a wireless link that does not interfere with the remaining mesh network's resources, e.g., a directional WiFi or WiMax link to a wireline gateway (see TFA for an example of directional WiFi gateways).

The user-specified cost of installing a capacity point (i.e., a physical wire or dedicated wireless connection) can be different for each location and I allow non-uniform capacities at each location. Further, I focus on a single-radio, single-channel backhaul and access tier architecture, and later extend the model to dual radio platforms that have a separate access and backhaul radio.

4.2.1 Gateway-Limited Fair Capacity

The proposed capacity calculation technique captures the impact of wireless contention on the utilization of the wireless medium in a computationally efficient manner. I consider access networks where all traffic to and from clients must traverse a gateway, making the gateways the bottlenecks in the network. Therefore, I focus on the performance of the gateway nodes, hence the gateway-limited fair capacity. The advantages of this model over previous, more general computational models [35] are 1)
exact computation in polynomial time (important for evaluating many possible local search operations) and 2) extension to local search algorithms by enabling tractable approximations which optimize over one of two components of capacity definition: route lengths or contention.

A key aspect of this technique for calculating capacity is to model the wireless interface of a gateway as alternating its time between transmitting to one-hop neighbors, receiving from one-hop neighbors, and deferring to other neighbors within contention range. The time a gateway spends deferring to ongoing transmissions in contention range reduces the gateway's available capacity. Therefore, I define the gateway-limited fair capacity as a function of the airtime utilization of the gateways, which depends on the routes used and amount of time the routes lead to a gateway deferring. In this definition, gateway capacity is significantly affected by fairness. For example, allocating all resources to one-hop flows and none to multi-hop flows will yield the greatest capacity but would be undesirable as large portions of the network would be non-functional. Consequently, I impose a per mesh node fairness constraint, requiring that each mesh node receive its fair share of the wireless airtime at the gateway nodes.

More formally, let $n$ be the total number of mesh nodes in the network, and $m$ the total number of links. Define $\mathcal{G}$ as the set of all potential gateway locations, which is a subset of $\mathcal{M}$, the set of all mesh nodes. Mesh node $i$ has a traffic demand
d[i] that represents the aggregate demand of all the end-clients associated with it. I represent the routes used by each mesh node to reach one or more gateways as a two-dimensional matrix \( R \), where \( R[i,j] \) indicates the amount of node \( i \)'s demand that traverses link \( j \). I designate \( src(i) \) as the access tier link for mesh node \( i \) and assign \( R[i,src(i)] = d[i] \). This formulation ensures fairness by requiring that \( \lambda d[i] \) units of mesh node \( i \)'s demand are served by gateways. The positive-valued \( \lambda \) parameter is uniform for all mesh nodes and therefore leads to weighted fair shares being enforced. I scale the demands with the \( \lambda \) parameter such that they are feasible, and then find the \( R \) matrix as solution to a transhipment problem optimizing capacity, potentially allowing multipath routing. I represent the contention caused by each link in a two-dimensional matrix \( I \), where \( I[i,j] \) indicates if link \( j \) is in contention range of node \( i \). The \( I \) matrix notation extends to links that, due to physical layer shadowing, only cause contention during a fraction of time.

The technique for calculating the amount of time a gateway is idle due to contention proceeds as follows. A link induces contention equal to the number of mesh nodes that cannot be actively transmitting or receiving when the link in question is active. Consequently, the total contention on a gateway depends on how many routes use the link and how much demand is routed over the link. I use contention as a simplification of interference, as I am concerned specifically with situations in which a node is forced to defer due to either concurrent transmission or interference.
I assume a perfect MAC protocol without unfairness or hidden terminal effects.

The total contention on a gateway node \( g \in G \) caused by link \( j \) is \( \sum_{i=1}^{n} R[i,j] \times I[g,j] \). The total contention on gateway \( g \), \( v_g \) is then given by:

\[
v_g = \sum_{j=1}^{m} \sum_{i=1}^{n} R[i,j] \times I[g,j]
\]  \hspace{1cm} (4.1)

The fair wireless capacity of a gateway is computed as follows. Gateway \( g \) services total demand \( s_g \), which is the sum of demands on all links incident to gateway \( g \), denoted by \( \text{link}(g) \):

\[
s_g = \sum_{i=1}^{n} \sum_{j \in \text{link}(g)} R[i,j]
\]  \hspace{1cm} (4.2)

Expressing the capacity of gateway \( g \) as the amount of wireless time \( v_g \) required to serve \( s_g \) units of time at the wired interface, \( u_g = s_g/v_g \). Thus, the total gateway-limited fair capacity is the sum of \( u_j \) terms for all \( j \in G \).

This sum is a lower bound of the actual gateway-limited capacity due to potential double-counting of links in contention with the gateway. This may occur if two links that contend with a specific gateway are not in contention with each other and can therefore be active simultaneously. In this case, the gateway is deferring to two links at once, whereas my calculation would count separately the defer time for both links.

4.2.2 Gateway Placement Problem

The gateway placement problem decides how best to place a fixed number of additional capacity points in an existing mesh network so as to maximize the overall
capacity improvement. It is formally defined as follows. Let G be a \((0, 1)\)-vector of size \(n\) that indicates whether a given mesh node \(i\) is a capacity point or not. On an operational mesh network, \(G[i] = 1\), for all \(i \in \mathcal{G}\). Let the monetary cost of installing a capacity point \(i\) be \(f[i]\) and the set \(\mathcal{G}\) represent the currently deployed capacity points I define the total cost, \(C(G)\), of installing new capacity points in the mesh network as:

\[
C(G) = \sum_{i \in \mathcal{G}} f[i] \times G[i].
\]

I express the gateway placement problem as maximizing the network capacity given a specified budget for adding capacity points. This formulation contrasts with previous work ([5] and [6]) which does not directly account for wireless contention effects or consider the need for upgrading existing mesh deployments.

The placement problem is difficult because it requires simultaneously solving three subproblems: 1) gateway selection, 2) client assignment to gateways, and 3) route selection. The approximation schemes presented in the next section use local search techniques to decouple these subproblems. The algorithms solve (2) and (3) together (i.e., a transhipment problem) in order to evaluate the effectiveness of all possible local operations and thereby choose operations that best solve (1).
4.3 Solving the Placement Problem

The gateway placement problem involves maximizing capacity directly, which can be expressed as an integer program (IP). I instead propose two local search based algorithms due to the following disadvantages of an IP: (i) an IP cannot solve the problem exactly in polynomial time, (ii) prior work has shown that a simplified version of the problem, capacitated facility location, has an unbounded integrality gap [38], and (iii) an IP is not suitable for online computation, e.g., it precludes the case of incrementally adding gateways without recomputing the locations of every gateway.

I therefore take an alternate approach of maximizing capacity with local search algorithms. I present two algorithms that optimize based on one of the two major components of gateway-limited fair capacity: the size of the routes in R or the impact of contention in I on mesh nodes. I first maximize capacity as a capacitated facility location problem with budget constraint and solving with minimum hop-count based local operations open() and close() and iterative capacity estimation. The second approach is to minimize average contention as an uncapacitated k-median problem, solved by local swap() operations which results in a polynomial-time approximation algorithm.

4.3.1 Solving by Maximizing Capacity

I first review the facility location problem and then describe how to map the gateway placement problem.
Facility Location Problem

The gateway placement problem is a generalization of the capacitated facility location problem [36], which is defined as follows. Let $\mathcal{M}$ be a set of customers, and $\mathcal{W}$ be a set of facilities. Each customer $i \in \mathcal{M}$ has a demand $d[i]$. Each facility $j \in \mathcal{W}$, has a maximum capacity $u[j]$ and a facility cost $f[j]$. The cost matrix $C[i,j]$ represents the cost of serving one unit of demand from customer $i$ by the facility $j$. A facility can satisfy a customer demand only if it is open. The facility location problem then is finding a set of facilities to open, $\mathcal{G}$, with minimum total cost:

$$\sum_{j \in \mathcal{G}} f[j] + \sum_{j \in \mathcal{G}, j \in \mathcal{M}} C[i,j] \times X[i,j]$$

where $X[i,j]$ denote the fraction of demand from customer $i$ served by facility $j$.

In my formulation of the gateway placement problem, the facilities map to capacity points and the customers correspond to mesh nodes. The key differences are:

- The wireless capacity of each gateway depends on nearby contention, which in turn depends on the placement of other gateways. Therefore, the capacities are not known \textit{a priori} because it would require knowledge of the final placement.

- Defining a cost function for serving mesh node $i$ by gateway $j$ does not preserve the triangle inequality. This cost is equal to the fair share of mesh node $i$ at gateway $j$. If the customer and facility cost metrics do not preserve the triangle inequality, no constant factor approximation algorithms are known.
Despite these differences, the local search algorithms developed for the facility location problem apply to the gateway placement problem (both differences are addressed in the adaptation of the algorithm, as described later).

**Local Search Operation**

I next describe the local search algorithm [36] for the facility location problem in the context of the gateway placement problem, highlighting modifications to account for gateway placement specifics. I denote $s$ as a node and $T$ as a set of nodes, which I describe how to find later in this section. The algorithm can do one of three operations to improve the solution: $add(s)$ installs a gateway at node $s$, $open(s, T)$ installs a gateway at node $s$ and removes gateways at all nodes in set $T$, and $close(s, T)$ removes the gateway at node $s$ and installs gateways at all nodes in set $T$.

Let the set of available gateway locations be $\mathcal{W}$, which is a site-specific subset of all mesh node locations. Let $\mathcal{G}$ represent the set of installed gateway locations throughout the execution of the algorithm, i.e., $G[i] = 1$, if $i \in \mathcal{G}$. The local search algorithm operates as follows. I start with an arbitrary valid gateway placement and perform one of the three operations, $add()$, $open()$, and $close()$, to improve the quality of the solution. To ensure the algorithm terminates in polynomial time, I require that each step lowers the cost by at least $c(S)/p(n, \epsilon)$, where $p(n, \epsilon)$ is a chosen polynomial in $n$ and $1/\epsilon$. Here, $\epsilon > 0$ indicates the error tolerance, and the algorithm's run time is polynomial in $1/\epsilon$. 
I now review in more detail each local search operation. Because all possible combinations for set $\mathcal{T}$ cannot be evaluated in polynomial time, the algorithm instead finds a good choice for the set $\mathcal{T}$ as the solution to a knapsack problem, where $\mathcal{T}$ is found as the set of items to put in the knapsack. The operations proceed as follows:

- $add(s)$ – For all non-gateway nodes $s$, evaluate the cost to open a gateway at $s \in \mathcal{W}$. This cost evaluation requires solving a transshipment problem to find optimal routing matrix $R$ for the set of all installed gateways in $\mathcal{G} \cup \{f\}$.

- $open(s, \mathcal{T})$ – Install gateway at node $s \in \mathcal{W}$ and remove gateways in set $\mathcal{T} \subseteq \mathcal{G} - \{f\}$, reassigning mesh nodes served by $\mathcal{T}$ to the gateway at $s$. Note that gateway $s$ could already have been installed with some unused capacity.

- $close(s, \mathcal{T})$ – Remove gateway $s \in \mathcal{G}$ and install a set of gateways $\mathcal{T} \subseteq \mathcal{W} - \{f\}$. Then reassign routes destined to $s$ to gateways in $\mathcal{T}$ without any effect on mesh nodes served by other gateways.

**Adapting MinHopCount**

I next describe the necessary modifications to allow the facility location algorithm to maximize capacity subject to a budget constraint and gateway capacities. I then describe the technique for iteratively estimating gateway capacity.

The proposed algorithm to maximize capacity is termed MinHopCount because it uses the minimum hop count to find good candidate local search operations. I then
apply the local search operation that results in the largest capacity increase. The hop count is a useful cost function in this problem because it is a first-order approximation of the capacity, i.e. it reduces the contention in Eq. 4.1 by reducing the value of \( R \) entries. Another important advantage of this metric is that it preserves the triangle inequality, which is necessary for provable bounds on the local search algorithm's performance.

I also add a budget constraint to the MinHopCount algorithm, making the problem solved a generalization of the capacitated \( k \)-median problem (more general because I allow all gateway costs, \( f[i] \), to be different). While there are no known constant factor approximation algorithms for the capacitated \( k \)-median problem, I show through evaluation that the algorithm performs close to optimal in realistic topologies (see Section 4.4).

The local search operations find a placement subject to gateway capacity constraints, but these gateway capacities are not known \textit{a priori} because they depend on the full gateway placement. As shown in Table 4.3.1, I use lower bound estimates for the gateway capacities, \( u[i] \), and iteratively update the gateway capacity lower bounds after successive runs of the local search algorithm. The algorithm terminates when the current sum of the lower bound capacity estimates \( u_{\text{cur}}[i] \) does not decrease by more than user-chosen parameter \( \phi \) from the previous iteration's estimate, \( u_{\text{prev}}[i] \). Intuitively, this approach finds the lower bound capacity of a near-optimal placement,
which is a tighter bound than the worst-case lower bounds of all placements. The run time of the MinHopCount algorithm is polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\phi}$.

4.3.2 Solving by Minimizing Contention

The second proposed local search algorithm, MinContention, finds the gateway placement that minimizes the average contention in the network. I first describe the $k$-median problem and review a local search algorithm using local $swap()$ operations. Second, I discuss how to map the gateway placement problem to this algorithm such that it finds the placement with the lowest average contention region size with a provable approximation ratio of $3 + \epsilon$.

The $k$-Median Problem

The $k$-median problem is a variant of the facility location problem where there are only a fixed number $k$ of facilities that can be opened. The objective is to minimize the cost of connecting all clients to a facility. I consider the uncapacitated version of the problem as there is currently no known constant factor algorithm for the capacitated $k$-median problem. In contrast, there is a local search algorithm for the uncapacitated $k$-median problem with a locality gap of $3 + 2/p$ [37], where the locality gap is the maximum difference between the worst local optimum and the global optimum and the parameter $p$ controls the number of gateways the algorithm considers for simultaneous swapping. This locality gap results in an approximation
ratio of $3 + \epsilon$. This local search algorithm is based on repeatedly swapping $p$ open gateways for $p$ unopen gateways until no swaps can improve the solution. A larger $p$ value leads to more accurate results but with exponential increase in running time.

The main idea of the MinContention algorithm is to install $k$ gateways to minimize the average contention on the mesh nodes, which is a function of which links contend with each node and how often those links are used in routes. As per the capacity definition in Section 4.2, the actual objective is to minimize the total contention on gateways, but this objective cannot be directly achieved because it requires knowing the full gateway placement to correctly assign link weights. I therefore solve the problem of minimizing the contention on all nodes as a means of approximating the gateway contentions. Note that a disadvantage of this algorithm over the previously discussed MinHopCount algorithm is that it requires identical gateway costs.

**Swap-based Local Search**

The MinContention algorithm is summarized in Table 4.3.2. The cost of a placement is the sum of the active link weights, which are each assigned to be the total number of mesh nodes in contention range of the link. Additionally, I scale the shortest paths' weight in proportion to the node's traffic demand. This allows for taking into account the client demands and potentially installing gateways nearer to mesh nodes with greater demand.
Triangle Inequality for Contention

In order for the above algorithm to have a provable locality gap of $3 + 2/p$, the link weights must obey the triangle inequality. As previously stated, the assigned link weight is the size of the union of the sets of nodes in contention range with each endpoint of the link, which preserves the triangle inequality. The proof of this is, as discussed in the previous chapter, due to the definition of contention as an edge weight. Summarizing the argument in Section 3.3, the contention on an edge is the union of the nodes which contend with either end point of the link. The key property of this union is that there can then be no shortcut through the graph with less contention because it will always at least include the contention due to both endpoints.
Let $\mathcal{M}$ be the set of all mesh nodes

Initialize $u[i]$ values to gateway wired capacities

Do {
    Start with arbitrary, valid solution $G$
    Do {
        Foreach $s \in \mathcal{M}$
        Find valid $add(s)$
        Find valid $open(s, T)$
        where $T$ is solution to knapsack problem with size = $u[s]$
        Find valid $close(s, T)$
        where $T$ is minimal covering knapsack with size = $u[s]$
        Calculate $\Delta$ cost for all valid operations
        Apply operation to $G$ with best $\Delta$ cost
    } while ($\Delta$ cost $\geq C(G)/p(n, \varepsilon)$)

Output $G$ as locally optimal solution

Calculate capacities $\hat{u}[i]$ of placement $G$

Update $u_{\text{cur}}[i]$ to new lower bound if $\hat{u}[i] < u_{\text{prev}}[i]$

} while ($\sum_{i=1}^{N} u_{\text{prev}}[i] - u_{\text{cur}}[i] \geq \phi$)

Output $G$ as solution

\begin{table}
\caption{Pseudocode for MinHopCount algorithm.}
\end{table}
Find a feasible starting placement $G$

Do {

Find all valid $\text{swap}(S, T)$

where $S$ is set of $p$ gateways to open

and $T$ is set of $p$ gateways to close

Calculate $\Delta$ cost for each operation

Apply $\text{swap}$ with largest positive $\Delta$ cost

} while ($\Delta$ cost $\geq \frac{O(G)}{p(n,c)}$)

Output $G$ as locally optimal solution

**Table 4.2** Pseudocode for MinContention algorithm.
4.4 Evaluation

This section examines the performance of the proposed placement algorithms. I first examine the proposed technique for capacity calculation with measurement data. I next study the algorithms on regular grid topologies and then real topologies that underlie three deployed mesh networks, and then study the characteristics of an optimal placement. Finally, I consider the joint placement of mesh nodes and capacity points.

4.4.1 Validating Capacity Calculation

This section demonstrates the ability of the proposed capacity calculation technique to accurately rank gateway placements from highest to lowest total capacity.

Validation via Simulations

First, I validate my proposed capacity calculation technique on a simulated topology, a $4 \times 4$ regular grid network, using the NS-2 network simulator. The mesh nodes are spaced 225 meters apart and use the default transmission and interference ranges (250 and 550 meters respectively). For this experiment, I assume the access tier is on a separate channel and therefore clients do not impact contention. The chosen gateway node sends a fully backlogged UDP flow to each mesh node in the network. I do not record the traffic destined for the gateway node because it does not traverse the backhaul tier and is instead limited only by the access tier capacity.
The objective of this experiment is to study gateway placement predictions using the gateway-limited model in comparison with simulation. Figure 4.1 depicts the model and simulation capacities, ranked in order based on the simulation results. The ranking matches closely for both, although the simulation results do not achieve the predicted capacity due to congestion and overhead effects. Note also that I only consider downlink traffic flows. In the uplink direction, the traffic experiences severe unfairness [39], violating the fairness requirement of the proposed capacity definition.

![Graph](image)

**Figure 4.1** Validation of model with ns-2 simulation, placing one gateway on 4x4 grid topology and ranking each of the possible 16 configurations.
Validating via Measurements

I next compare my calculation technique with measured throughput data from the operational TFA mesh network, in order to show that a better placement as per the capacity calculation is also a better placement as per measurements. TFA is a multi-tier mesh network providing Internet access in a densely populated, single-family residential, urban neighborhood with 18 deployed mesh nodes [10]. In the topology, two mesh nodes are connected if their link is on average usable at greater than 1 Mbps.

At the time of these experiments, the TFA network, featured two capacity points: gateway GW-A is a true wireline gateway and gateway GW-B is connected to GW-A via a directional link. I measure three different capacity point configurations: GW-A only, GW-B only, and both GW-A and GW-B. I observe the network during weekday peak hours for each of the three configurations, each measured on a subsequent weekday. For example, on one day, the directional link was disabled, making GW-B a non-gateway mesh node. The measured throughput is the peak rate (in Mbps) of data flow between the TFA network and the Internet over the observed time period. The traffic measured is the naturally occurring usage of the network. The throughput of the network with both gateways peaks at 2.2 Mbps. The GW-A configuration has a peak throughput of 1.46 Mbps and the GW-B configuration peaks at 610 Kbps. Using the technique in Section 4.2, I calculate gateway-limited fair capacity of the
GW-A-only configuration as 1.5 Mbps, the GW-B-only configuration as 1.35 Mbps, and both gateways together as 1.75 Mbps. The gateway-limited fair capacity predicts the correct ranking, with the GW-A-only configuration achieving 11% greater capacity than the GW-B-only configuration. The actual throughput values are lower than the calculated capacity due to several factors not included such as control overhead, MAC unfairness, and non-backlogged traffic sources.

4.4.2 Performance of Placement Algorithms

This section studies the performance of MinHopCount and MinContention algorithms presented in Sections 4.3.1 and 4.3.2 on grid-based and realistic topologies. For all experiments, I consider an 802.11b system with the single-link wireless throughput assumed to be 6 Mbps. All mesh node locations are fixed and gateways can be installed on any mesh node. I compare the algorithms against a greedy placement strategy which repeatedly places the gateway that leads to the largest reduction in average path length.

Regular Grid Topology

I first examine the placement algorithms on a 7 × 7 regular grid topology. A mesh node communicates directly with at most 4 neighbors and contends with all two-hop neighbors. The only significant distinction between nodes are those that are on the borders. Figure 4.2 shows the performance of the greedy, MinHopCount, and
MinContention algorithms as a function of how many new capacity points are added.

For a network of this size, the results also include the optimal placements found using brute force search.

![Figure 4.2](image-url) Capacity of placements arrived at by algorithms on a square grid topology with 49 mesh nodes.

For adding between three and six gateways, the MinHopCount and MinContention algorithms find placements with capacities at least 86% and 77% of the optimal respectively, while the greedy placement is at least 72% of the optimal. The MinHopCount algorithm performs better in this regular topology because contention is uniform, leading to fewer or no situations where a greater hop count leads to better capacity. I also find that the each algorithm sometimes performs slightly worse with
more gateways due to the fact that the hop count and contention metrics they use are only a first order approximation of the mesh capacity. Note that, in this topology, the marginal benefit of each new gateway decreases due to the increasing level of contention between gateways. This effect is significant as gateways serve the most traffic and therefore cause more contention than other mesh nodes.

**Real-World Topologies**

I next evaluate the placement algorithms on the topologies of three currently deployed mesh networks: TFA, Chaska, and Google. These topologies present a new challenge in that the connectivity and contention matrices are no longer uniform for each mesh node. In these topologies, the local search algorithms have greater gain over greedy heuristics than in grid topologies because the irregular contention leads to situations where longer routes result in higher capacity. For each topology, I fix a number of already installed gateways and focus on upgrading with new gateways.

For the TFA topology, I am able to directly measure the signal strength between each pair of nodes. The topology is then a combination of this information with empirically measured communication and contention thresholds. For the Chaska and Google topologies, I estimate the connectivity information with AP coordinates and manufacturer's information, introducing possible errors. While the true connectivity matrix is not observable externally, I assume a link exists if the physical distance is less than 200 meters.
The first deployed topology corresponds to the 195 node Chaska topology [40]. I begin with four known gateways and place additional capacity points in the network using the greedy, MinHopCount, and MinContention algorithms, plotting the results in Figure 4.3. Optimal does not appear here as the network size prohibits exhaustive search. The MinContention algorithm typically performs the best, up to 64% better than the greedy placement, because the local search improves upon previous choices it made and the algorithm considers the amount of contention caused by a path and not just the path length. For this topology, the MinHopCount performs up to 30% better than greedy, but its performance becomes similar to greedy beyond 15 gateways. The MinHopCount’s capacity estimates, u[i], degrade for more than 15 gateways for this topology, and it is the main reason that its performance becomes similar to greedy.

The second deployed topology considered in Figure 4.3 is the 447-node Google Mountain View network. In this network, I consider the current configuration of 59 gateways and use the proposed algorithms to determine upgrade locations. MinContention outperforms greedy by up to 8%, though not with small budgets. Conversely, MinHopCount performs best with small budgets, but is approximately 10% lower capacity than greedy when considering larger budgets. This is a result of the simple capacity estimation strategy, which does not take into account contention between gateways. Note that the topology estimation results in a conservative and regular contention pattern in this topology.
The third deployed topology considered is the TFA network expansion, consisting of the currently deployed 18 nodes and 35 planned nodes. The current topology features two capacity points: one wired gateway and one directional antenna connection. Figure 4.4 presents the results of adding a small number of gateways to the projected TFA topology while holding fixed the current two gateways. Also included are the optimal values found via exhaustive search.

The local search algorithms closely approximate the optimal solution for the addition of up to 3 gateways; MinHopCount and MinContention solutions are within 97% and 96% of the optimal. As the budget increases, the solutions decline to as low
as 80% of the optimal, with MinHopCount declining more. Greedy performs worst with small budgets, but improves as the marginal gain of additional gateways declines and allows the greedy to make up for early suboptimal choices. The MinHopCount algorithm performs worse with five gateways than with four due to the fact that iterative capacity estimation does not directly take into account inter-gateway contention and hence MinHopCount does not perform as well when gateways contend with each other. In other words, it conservatively chooses longer paths so as to ensure that gateway capacity constraints are not violated.

In summary, I found that the proposed local search algorithms significantly out-
perform a greedy algorithm, by up to 64%, and this gain is more pronounced on irregular topologies. For small budgets, the algorithms achieve very close (≥ 97%) to optimal capacities and for larger budgets, MinContention performs best.

4.4.3 TFA Placements Case Study

I next study in greater detail the TFA network using an exhaustive study of all possible placements. I find that while the best placements have similar configurations, i.e. roughly the same gateway locations, there is a large capacity gap between near-optimal placements and the optimal. In other words, the optimal placement has significantly higher capacity than a near-optimal placement, demonstrating the need for good approximation algorithms. Further, the configuration of gateways in the optimal placement is similar to a near-optimal placement, indicating the applicability of local search operations for finding optimal placements. I consider the case of adding four additional capacity points in the projected TFA topology.

Distribution of Placement Quality

Figure 4.5 presents a histogram of all possible ways to install four additional capacity points. Four candidate locations have been chosen based on availability of structures to mount antennas and I compare this manual placement with the proposed algorithms. To understand the space of possible placements, Figure 4.5 is a histogram of the capacities resulting from all possible gateway placements, found
via exhaustive search. The average placement results in a capacity of 7.7 Mbps with standard deviation of 1.2 Mbps and the optimal placement is 11.7 Mbps.

![Histogram of all placements of 4 new gateways in the projected TFA topology, along with the capacities found by the evaluated algorithms.](image)

**Figure 4.5** Histogram of all placements of 4 new gateways in the projected TFA topology, along with the capacities found by the evaluated algorithms.

The difference in capacity between the mean placement and the optimal placement is a factor 1.7×, indicating the need for a good approximation algorithm. The MinHopCount and MinContention algorithms achieve 85% and 79% of the optimal configuration respectively, whereas the greedy placement achieves less than 60% of the optimal. Also, MinContention with \( p = 1 \) gives the same capacity as with \( p = 4 \) (maximum \( p \) possible in this case because \( p \leq k \)).

Figure 4.6 plots the percentage of all possible placements with higher capacity.
Figure 4.6  Percentage of the total number of possible placements that result in a better capacity than the placement found by the proposed algorithms.

than the placement arrived at by the considered algorithms for this TFA scenario. MinHopCount achieves 85% of the optimal capacity, but only 0.1% of all possible placements result in higher capacity. For MinContention and greedy, approximately 1% and 15% of all placements result in higher capacity respectively. These results demonstrate the importance of a good approximation algorithm as there are large capacity gains due to finding better placements from among the top 0.1% of all placements.
Characterizing Similarity of Placements

Next, this section examines the characteristics of the solutions found by the local search algorithms in comparison to the optimal placement. I define a simple metric to capture the amount of similarity between any two gateway placements: the hop distance between the gateways in the two placements. The distance is calculated as the minimum hop cost to move the gateways in one placement to match the gateways in the other. This is equivalent to a transshipment problem where the demands are the capacities of the gateways in one placement, and the capacities correspond to the gateways in the second placement.

![Figure 4.7](image)

**Figure 4.7** Scatter plot for all possible gateway placements in the TFA-proj topology with 4 new gateways.
Figure 4.7 plots the ranges of capacities obtained when adding four additional gateways to the TFA network as a function of their distance from the optimal placement using this metric. I find that there is a strong correlation between distance from the optimal placement and the capacity with a correlation coefficient of $-0.807$. In other words, the higher capacity placements are most likely to be similar in configuration to the optimal placement. A carefully designed local search algorithm can take advantage of this similarity to find the optimal placement. In the example of adding four gateways, the distance between the second-best placement and the optimal placement is four hops. This means that the second-best placement is a minor perturbation of the best placement, and therefore the optimal placement can be found from the second-best placement with a local operation that moves gateways a combined total of four hops (not four hops per gateway). Both of the proposed algorithms find placements within six hops from the optimal.

4.4.4 Jointly Placing Mesh Nodes and Gateways

The prior evaluation of gateway algorithms assumes that the mesh node placement is fixed, and I now consider both the proposed mesh node and gateway placement algorithms in conjunction. Mesh node locations impact the calculated gateway-limited fair capacity by changing which links contend with each other and providing to the routing protocol choices that result in lower contention on the gateways and therefore higher capacity.
Figure 4.8  Comparison of two-stage placement and joint placement of mesh nodes and gateways in the TFA neighborhood topology.

Figure 4.8 compares the resulting capacity on the TFA topology using 1) a two-stage approach of mesh node placement and then gateway placement and 2) a joint algorithm that places both resources together. Specifically, I augment the MinContention algorithm to also greedily add new mesh node locations or swap pairs of existing mesh nodes with unchosen potential locations. The joint algorithm resulting in 5 - 15% more capacity, due to moving mesh nodes that are two hops from a gateway out of contention range of the gateway. The largest gains arise in topologies where gateways contend with each other (dense gateway placements) because the reduction in contention is compounded by the fact that two gateways in contention each can be
active at most half of the time.
4.5 Related Work

A gateway placement algorithm using a greedy heuristic has been presented [5] to serve neighborhood networks, as well as a local search algorithm [6] for minimizing a combined cost and hop count metric. These thesis differs from prior approaches because it 1) incorporates wireless contention, 2) considers deployed city-wide mesh topologies, and 3) presents two local search approximation algorithms, one of which has provable constant-factor approximation ratio. Others developed general techniques to calculate network capacities with interference [35] and incorporating multiple radios and channels [41]. These techniques require solving linear and mixed integer programs to find upper and lower capacity bounds. In contrast, I present a simple technique for exactly calculating gateway-limited fair capacity in polynomial time.

The capacity of hybrid wired and wireless networks has been studied in [34], though this study only provides asymptotic bounds and does not address the gateway placement problem. For regular topologies, [42] studies the impact of gateway density on network capacity and presents techniques to calculate connectivity to gateways.

The algorithms proposed in this thesis build upon solutions to the related capacitated facility location problem and uncapacitated $k$-median problem. Constant-factor approximations are known to exist for these problems using both local search [37, 36] and LP relaxation methods [43]. The proposed algorithms can be improved with more
sophisticated local search techniques [44, 45] that achieve better approximation ratios. For the closely related capacitated $k$-median problem, there is a constant-factor algorithm with up to $50 \times$ violation of capacity constraints [46], making the algorithm too inaccurate for practical purposes. Finally, prior work has also adapted these algorithms in the networking domain for placement of content server replicas [47].
4.6 Summary

This dissertation studies the gateway placement problem, first introducing a technique to efficiently compute gateway-limited fair mesh capacity as a function of the contention at each gateway. We then present two gateway placement algorithms adapted from local search heuristics for related facility location problems with provable performance guarantees. The MinHopCount algorithm adapts a local search algorithm for the capacitated facility location problem and minimizes the average wireless hop count for all paths in the network, iteratively estimating the gateways' wireless capacities. The MinContention algorithm is adapted from a solution to the uncapacitated $k$-median problem and minimizes the average contention region size within a provable approximation ratio of $3 + \epsilon$. MinHopCount is more general and can handle non-uniform gateway costs, while MinContention is able to provide better performance guarantees. Our numerical results on three real topologies show that our algorithms outperform a greedy heuristic and achieve close to the optimal capacity. Further, we show that near-optimal solutions have similar gateway configuration as the optimal, but the difference in capacity is large, which supports the use of local search operations on near-optimal placements.
Chapter 5
Measurement Assessment of Deployed Networks
5.1 Introduction

To evaluate, expand, or improve performance in a deployed network, a network operator must first assess the current spatial performance of the network. This chapter formulates the wireless network assessment problem as a problem of identifying metric regions, i.e., identifying locations in the network where the given performance metric meets a conformance threshold. Existing assessment strategies either require exhaustive measurements [7] or use detailed physical-layer object descriptions to precisely estimate propagation characteristics [8]. Unfortunately, these approaches are expensive and often impractical, especially for incremental network upgrades.

In this thesis chapter, I present a general framework to assess the spatial performance of a deployed mesh network using a constrained number of measurements. I estimate metric regions by coupling the use of coarse-grained terrain maps with the construction of virtual sectors of differing radii overlaid on the physical topology. I then validate the framework's accuracy and study two deployed urban mesh networks using measurement sets from approximately 30,000 client locations in each network. In particular, the contributions are as follows.

First, I devise a mesh network assessment framework that divides each node's metric region into a number of virtual sectors. I use a two-stage process to first estimate the metric sector boundaries (radii) and then to refine each boundary estimate through the selection of a small number of measurements. For coverage, the difficulty
in estimating a metric region is due to complex and highly variable interactions with the physical environment, e.g., see [48]. Thus, I use the geometry of the terrain obtained from publicly available digital maps to account for differences in propagation characteristics among regions. Moreover, I refine the estimated sector boundaries with a small number of measurements guided by a push/pull heuristic that selects measurement locations and adjusts the estimated boundary. By assuming a monotonic decay in performance with distance, one can model a metric region as a single, connected, multi-sector, multi-radii region. I show that despite some monotonicity violations, the presented framework obtains high characterization accuracy.

I then validate the framework and evaluate two operational networks by performing an extensive set of client measurements from two currently deployed wireless mesh networks, GoogleWiFi and TFA. I show that for a given accuracy in describing metric regions, the framework requires two to five times fewer measurements than a grid sampling strategy. I compare the boundary refinement heuristic to a simplified ray-tracing approach, and show that the proposed heuristic is more robust to monotonicity violations and obtains better accuracy. Using the assessment framework, I find that the TFA network is deployed with sufficient density so that coverage holes occur only on the network edges. However, for the GoogleWiFi network, the frequency of coverage holes is much less dependent on deployment density, although the size of the holes decreases for higher node densities. This points to a key challenge
of covering holes with additional nodes: a large number of additional nodes would be required to eliminate numerous small coverage holes in which half have radius of less than 10 meters. Lastly, I investigate client association policies and find a 20% loss in client throughput due to uneven spacing of mesh nodes.

The rest of this chapter is organized as follows. Section 5.2 defines the assessment problem and proposed framework. Section 5.3 presents estimation, sectorization, and measurement-based refinement algorithms. Section 5.4 validates the accuracy of the assessment framework with real data sets and Section 5.5 evaluates two studied deployments. Section 5.6 discusses related work and finally, Section 5.7 summarizes.
5.2 Assessment Framework

This section formally defines the network assessment problem and then sets up the required notations and definitions for the estimation and refinement framework.

5.2.1 Problem Definition

The problem I address is how to accurately characterize the coverage of a network with a small number of measurements. Generally, I characterize a network by identifying specific areas in the network where the measured value of a given metric (e.g., signal strength) exceeds a given threshold. Therefore, a good characterization scheme is one with a high accuracy for identifying unmeasured locations as above or below the given threshold. This formulation addresses scenarios such as: 1) a network operator wishing to identify dead-spots in order to add nodes and improve performance, or 2) a municipality wishing to determine if a deployed network conforms to contractual performance requirements. I next formally define the assessment problem with a constraint on the number of measurements allowed.

I consider a terrain $\mathcal{T}$, which consists of a continuous space of points, $p \in \mathcal{T}$, on a 2-d Cartesian plane. Similarly, I define the set of mesh nodes $\mathcal{N}$, where each node $n \in \mathcal{N}$ is defined by a coordinate pair in 2-d space. Let $M$ represent a specific performance metric; the study focuses on an SNR-based coverage metric, but also includes modulation rate and redundancy. For each point $p$, I define $M(p)$ as the measurable value of metric $M$ at point $p$. Measurement cost is assumed to be identical
for all points \( p \in T \), but not for all metrics.

I begin by characterizing a single point with respect to a given metric \( M \) and given threshold, \( \theta_M \), which represents the metric's performance cutoff. A point \( p \) satisfies metric \( M \) if \( M(p) \geq \theta_M \). Characterization is then defined as correctly predicting if a location satisfies metric \( M \). For metric \( M \), a mesh node's metric region is the set of all points \( p \in T \) such that \( M(p) \geq \theta_M \). A mesh network's metric region is the union of all mesh node metric regions in set \( \mathcal{N} \).

The role of measurements in the characterization of a network is to gain additional knowledge with which to increase the accuracy of predicting the value of \( M \) at an unknown location. In order to limit the measurement expense of the assessment study, I add a constraint that limits the total number of measurements. Stated as an optimization problem, assessment seeks to maximize the characterization accuracy over a terrain \( T \) subject to a constraint on the total number of measurements taken.

I consider both versions of the problem which characterize either the metric region of a single node or of an entire network.

A key challenge for signal strength based metrics is that physical-layer transmissions do not propagate uniformly at all angles from a mesh node and signal strength does not monotonically decrease with distance. Further, there is no known practical way to \textit{a priori} characterize the large changes in signal strength over short distances.
5.2.2 Metric Definitions

I first define three performance metrics for coverage, modulation rate, and redundancy. The coverage metric is based on the received signal-to-noise ratio (SNR), labeled $P_{dB}(p, n)$, at a client point $p$ from node $n$. A conformance threshold, $\theta_c$, indicates the minimum acceptable SNR.

**Definition 1** Consider a terrain $T$, a location $p$, and a mesh node $n$ in $T$. The location $p$ is covered by $n$ if the received SNR at $p$ with respect to $n$, $P_{dB}(p, n) \geq \theta_c$. The coverage region of $n$ is the set of all points in $T$ covered by $n$.

The second metric is modulation rate, which captures the expected value of the physical-layer modulation rate in use at a given location. This value is a function of SNR and the rate selection protocol used, e.g., Auto-Rate Fallback (ARF).

**Definition 2** Let $n$ be a mesh node and $p$ be a client location in a terrain $T$. The modulation rate of $p$ with respect to $n$ is the expected physical layer modulation rate in use. The modulation-rate region of $n$ is the set of all points in $T$ with expected modulation rate at least threshold $\theta_r$.

I now define the coverage redundancy metric, which is based directly on the coverage metric and is the number of mesh nodes which cover a given point.

**Definition 3** The redundancy of a location $p$ in a terrain $T$ is the number of mesh
nodes that cover p. The k-redundancy region of T is the set of all points in T with redundancy k or greater.

5.2.3 Metric Sector Framework

The proposed assessment framework uses terrain information to divide the mesh node metric region into virtual sectors of varying angular widths and radii. To accurately characterize the network's diverse propagation environment, I independently estimate metric sector angles and boundaries.

More formally, a metric sector of mesh node n is a sector of the circle centered at n contained between angles \( \phi_1 \) and \( \phi_2 \). I consider monotonic performance metrics defined as follows. Let the function \( d(p_1, p_2) \) denote the distance between points \( p_1 \) and \( p_2 \) in a terrain, then:

**Definition 4** Let T be a terrain and M be a metric. M is monotonic in T if for every mesh node n in T, for any ray R emanating from n and for any two points \( p_1 \) and \( p_2 \) on R, if \( d(p_1, n) < d(p_2, n) \), then \( M(p_1) \geq M(p_2) \).

While I assume performance measures such as signal strength decay monotonically for each ray, the use of multiple sectors with different radii does not require monotonicity among rays nor among sectors. For example, a far away signal strength can be greater than that of a closer distance provided that the two points are on rays having different angle from the originating node.
I assume that this monotonicity property is satisfied for coverage and show later in Figure 5.8 that the coverage metric mostly satisfies this property. The modulation rate metric also satisfies monotonicity, whereas the redundancy metric does not.

Let the *boundary* of a metric sector be the arc segment between angles $\phi_1$ and $\phi_2$, which defines the sector's border at radial distance $r$ from the mesh node. With this definition, I characterize a monotonic metric at an unknown location based on whether it is inside the metric boundary or not. The disjoint union of all metric sectors and sector boundaries defines the metric region. Note that the region boundary is non-uniform as it depends on the environment specifics in the region, and is different for each performance metric.

Thus, the proposed framework overlays a sector-based structure on the assessment problem. The objective to maximize predictive accuracy translates to minimizing the difference between the estimated and true metric boundary. The framework provides three types of variables to optimize on a *per-node* basis: 1) the number of sectors, 2) each sector's boundaries, $\phi_1$ and $\phi_2$, and 3) the boundary distance $r$ for each sector. The optimal solution is approached as the number of sectors goes to infinity, allowing the boundary to vary over smaller and smaller angles. In practice, I employ a small number of sectors because there is significant correlation over moderate angular distances, and the grouped boundary allows refinement with few measurements per sector, increasing overall accuracy.
Figure 5.1  Example metric sectorization and boundaries for an example GoogleWiFi mesh node's coverage region.

Figure 5.1 depicts an example of the framework's operation with a mesh node in the center of the figure, six virtual sectors displayed, and the estimated sector boundaries. In the next section, I present techniques for choosing the sector borders $\phi$ and boundary distances $r$, in order to heuristically improve the selection of boundaries.
5.3 Estimation and Refinement

In this section, I describe the proposed estimation techniques, including an estimator for coverage that exploits terrain information from digital maps. I then show how to use the estimates to drive sectorization so that chosen sectors have an approximately uniform boundary throughout. Lastly, I present an online heuristic to choose measurement locations in order to refine the metric sector boundary.

5.3.1 Performance Metric Estimation

I first present a coverage estimator which exploits terrain information to improve accuracy. I also introduce simple estimators for the modulation rate and redundancy metrics. Both coverage and modulation rate satisfy monotonicity, while redundancy does not, although it is calculated as a function of the coverage metric.

For coverage estimation, the environment has an average propagation environment (path loss) throughout. Yet, specific areas exhibit different propagation behavior due to different terrain (e.g., streets vs. buildings). Thus, an antenna's transmission not only experiences different attenuation at each angle, but each ray also faces varying attenuation as it moves away from the source. To address this uncertainty, the key technique is to couple terrain maps with measurements in order to better estimate SNR at a point. This is accomplished by calculating an average path loss for the entire network, and then for each measurement pair, I use the terrain information to estimate the shadowing, i.e., the deviation (in dB) from the average path loss. I next
describe the terrain information and then the estimation equations.

Terrain features encompass any type of physical area of the input map, such as buildings, fields, or trees, all of which are approximated with polygons. Figure 5.2 shows the publicly available digital map that I use to extract the GoogleWiFi terrain feature information. The TFA terrain map is not shown, but is similar. The number of different feature types and resolution of the terrain features determines the amount of information gained from the map, and is dependent on how the map processing algorithm groups similar features. Edge-detection image processing algorithms can be used to input satellite and city maps [49]. The maps used in the evaluation show zoning information and so a simple heuristic algorithm suffices to perfectly identify all terrain features. The output of the map processing algorithm is the set of polygons representing the terrain features. I then use training measurements to assign attenuation weights, $C_f$, to each feature type to indicate the feature's impact on pathloss estimation. Note that the studied networks feature homogeneous antenna heights and the usage of 2-d maps.

I estimate coverage using the standard log-normal path loss equation with shadowing [7]. The key technique is to use terrain features to estimate the shadowing value for each individual link. Shadowing accounts for the random variations in signal strength between node and client pairs at the same distance $d(n,p)$, which are due to differences in the scattering and attenuation environment and is usually
Figure 5.2  Terrain map of Mountain View, CA, for the measurement study area.

represented as a zero-mean Gaussian random variable [7].

Therefore, instead of estimating based only on average path loss, I also define a
terrain-informed shadowing estimator, \( \beta(n, p) \), to capture the specific path's deviation
(higher or lower) from the average path loss. Recall that the received power \( P_{dB} \) is
a function of the measured power, \( P_0 \), at reference distance \( d_0 \), and the average path
loss exponent \( \alpha \). The estimated received SNR is then:

\[
P_{dB}(p, n) = P_0 - 10\alpha \log \frac{d(n, p)}{d_0} + \beta(n, p) \tag{5.1}
\]
The terrain-informed estimator, $\beta(n, p)$, depends on a) the terrain features in $T$ that lie along the ray between the mesh node $n$ and point $p$, b) the width of this ray's intersection with each feature, and c) the feature type and weight, $C_f$. Specifically, $\beta(n, p)$ is defined as the sum of each intervening feature's impact on pathloss:

$$\beta(n, p) = \sum_{f \in F} C_f \times w(n, p, f)$$

(5.2)

where $F$ is set of all features in the terrain $T$, $C_f$ is the weight of a feature (attenuation in dB per unit distance), and $w(n, p, f)$ is the intersection width of the ray between $n$ and $p$ on the terrain feature $f$. In other words, each terrain feature that a link intersects either adds or subtracts from the value of the estimated pathloss, as a function of the feature weight $C_f$.

The $\alpha$ and $C_f$ terms above must be determined with some measurement overhead for each network. Training measurement locations are chosen randomly throughout the terrain, where each link intersects a subset of the terrain features in question. The training measurements must pass through a representative set of terrain features to capture each feature's effect on pathloss. In other words, I take measurements driving around the edges of terrain features, as opposed to measurements within each feature. The measured SNR values and measurement distances, in combination with Equations (5.1) and (5.2), then lead to a system of equations with the parameters as unknowns. I use minimum-mean squared error fitting to choose values of $\alpha$ and $C_f$ which best fit the measurements and equations. Section 5.4 studies the num-
ber of measurements needed per feature type, found to be between 10 and 20 for high accuracy estimation. My approach for incorporating small-scale terrain features builds upon empirical models for outdoor path loss prediction [50] in macrocells with adjustments for terrain environments.

The modulation rate estimator builds upon the coverage metric as follows. The constant $C_r$ maps SNR to an expected modulation rate choice, $T(n, p)$, from the set of possible physical layer modulation rates as: $T(n, p) = C_r \times P_{dB}(p, n)$, where $C_r$ is dependent on the interface technology in use. Finally, estimation for redundancy derives directly from the coverage estimation discussed previously.

5.3.2 Estimating Monotonic Metric Regions

The objective of the estimation algorithm is two-fold: to choose sector locations (angles $\phi_1$ and $\phi_2$) and estimate the metric boundary distance of each sector. Because the total number of measurements needed is a function of how many sectors are used, the algorithm merges boundary sections to reduce the number of sectors considered to the desired number and to output sectors with approximately uniform propagation throughout. The only measurements required for this algorithm are training measurements for the metric estimation function parameters, e.g., values of pathloss exponent $\alpha$.

Algorithm Estimate-Mono-Metric-Region (terrain $T$, mesh nodes $\mathcal{N}$, metric $M$)
1. For every mesh node $n \in \mathcal{N}$, do Steps 2 through 6.

2. Pick a set of rays at uniformly spaced angles from the mesh node. Call this set $R$, where the number of rays is chosen to be significantly larger (e.g., 10×) than the desired number of final output sectors.

3. For each ray in set $R$, a) traverse the ray along the terrain map identifying terrain features and the respective type and attenuation; b) estimate the value of the metric $M$ using a metric estimator to identify the boundary point $x$ on the ray, where $M(x) = \theta_M$; and c) connect the boundary points $x$ on each ray to identify the estimated boundary of the metric sector.

4. Create a mapping from each ray's angular position to the estimated metric boundary distance, $d(n, x)$.

5. Curve fit a step function to the above mapping, minimizing the mean-squared error between the estimated boundary distance and step function approximation. The number of steps corresponds to the number of allowed sectors, the height of each step is the boundary distance of each sector, and the cutoff points of each step are the sector border angles $\phi_1$ and $\phi_2$.

6. Output set of sectors with borders defined by step function cutoff points.

Figure 5.3 shows an example of the estimation algorithm output. I divide the
region surrounding the mesh node into 360 sectors with equally spaced rays (one per sector) and estimate the boundary distance of each sector. Since estimation requires only a constant number of training measurements, the number of rays chosen in this step does not increase the measurement budget. I then merge the sectors to result in the sectorized ranges also plotted (step function). Also, included are the measured ranges for this mesh node for the angles where data points are available.

![Graph showing estimated and measured ranges](image)

**Figure 5.3** Example of sectorization process in algorithm Estimate-Mono-Metric-Region for a GoogleWiFi mesh node. As a function of angle from the mesh node, the plot shows the estimated ranges, the sectorized estimate, and the measured ranges.

### 5.3.3 Estimated Boundary Refinement

I now describe the algorithm to choose measurement locations in order to refine the boundary estimate of each sector. The refinement is challenging due to mono-
tonicity violations and a noisy boundary. Therefore, I present a push/pull refinement heuristic, which is robust to these challenges by keeping little state in order to recover from anomalous measurements. Generally, the algorithm measures at the estimated boundary and then either pulls or pushes the estimated boundary nearer or farther from the mesh node based on the measurement result. The algorithm terminates when a boundary is found or the measurement budget per sector is exceeded. Algorithm

*Refine-Estimate* (terrain $T$, wireless nodes $\mathcal{N}$, sectors, metric $M$)

1. For each mesh node $n \in \mathcal{N}$, do Steps 2 through 6.

2. For each sector of mesh node $n$, do Steps 3 through 5.

3. From the location of $n$, draw one bisecting ray through each sector and identify point $x$, where the boundary intersects the ray.

4. Perform heuristic boundary refinement. While per-sector measurement budget not exceeded, take one measurement as close as possible to the estimated boundary $x$. If measurement is in metric region, move boundary point $x$ away from mesh node by constant distance, and vice-versa if measurement is outside of metric region. Stop if measurement is within tolerance (e.g., $\pm 3$ dB) of threshold value. Label the resulting boundary point on the ray as $z$.

5. Draw arc through $z$ to identify the refined boundary estimate for the metric sector.
6. Merge all the metric sectors with revised boundaries to get the refined estimate of the metric region of \( n \).

**Algorithm Properties.** By limiting the number of measurements per sector in the algorithm Refine-Estimate and limiting the number of sectors in the algorithm Estimate-Mono-Metric-Region, I ensure an upper bound on the total number of measurements taken, which is the product of the number of mesh nodes, the number of sectors per mesh node, and the maximum number of measurements per sector. Measurements are required only for 1) training measurements to estimate parameters \( \alpha \) and \( C_f \) and 2) boundary refinement measurements in step 4 of algorithm Refine-Estimate. There are two reasons for the actual number of measurements to be less than this bound: a) boundary refinement requires fewer measurements, and b) overlapping mesh node regions allow a measurement to be taken for multiple mesh nodes at one time.

### 5.3.4 Non-Monotonic Metrics

The proposed estimation algorithm assumes a connected metric region and thus a single metric boundary, derived from the monotonicity property. I extend the assessment framework to non-connected regions by considering only pairwise estimation instead of boundaries. The key aspect that allows us to estimate a disconnected region is a positive (additive) value for \( C_f \) terms in Equation (5.2). I later show in Section 5.4 that pairwise estimation is initially more accurate, but does not gain as
much from additional measurements.

A simple ray-tracing method for improving estimates with measurements involves localized refinement of the $\beta$ term in the estimator Equation (5.1). Instead of assigning values of $C_f$ based on global terrain features, I estimate the $C_f$ terms for only those features within a sector. I take uniformly distributed measurements per sector in order to refine $C_f$ values. This simple ray-tracing algorithm has the benefit of not assuming a connected region.
5.4 Framework Validation

I now validate the proposed framework using measurements from the GoogleWiFi and TFA networks with three performance metrics: coverage, modulation rate, and redundancy. I first introduce the network architecture and my measurement methodology. For each metric, I evaluate the accuracy and measurement overhead of the proposed algorithms. For the coverage metric, I also discuss the sources of inaccuracy and the frequency of monotonicity violations. The remaining metrics build upon coverage regions and I evaluate how accurately they can be estimated with the assessment framework.

5.4.1 Framework Validation Methodology

The measurement study consists of measurements from approximately 35,000 locations in the GoogleWiFi network and 29,000 locations in the TFA network.* This section validates the proposed framework’s accuracy using only small (100s of measurements) subsets of the data, which are then used to predict the full data set. Each coverage measurement corresponds with a GPS location reading. For the modulation rate metric, measurement pairs consist of the current SNR and modulation rate, and I extrapolate these measurements to a full set (all locations) of modulation rate measurements based on the probability distribution at each SNR value. The coverage threshold is set at $\theta_c = 25$ dB as this value allowed download throughputs of

---
*All measurement data available for download at http://tfa.rice.edu/measurements/
approximately 1 Mbps in both networks.

**Removing Measurement Bias**

Each coverage measurement point is a 4-tuple with \( x, y \) GPS coordinates, a node identifier, and an SNR value. I measure locations throughout all streets in the studied area, i.e., wardriving, as well as measuring in parking lots and driveways where possible. As the studied networks are not intended for pervasive indoor access, this work focuses on outdoor measurements.

I calculate the coverage of a network as the fraction of locations with measured signal strength above threshold \( \theta_c \). Because the error of the GPS positioning is 3 meters, I consider a single location as covered if there exist any measurements above \( \theta_c \) within a 3 meter radius. *Predictive accuracy* is then defined as the fraction of locations that the assessment framework correctly predicts as covered or not.

To account for an uneven spatial distribution in the measurements, I generate a set of 2-d sample points from a spatial Poisson process with intensity of 10,000 samples per km\(^2\). The evaluation then considers only sample points that are within 3 meters (the GPS accuracy) of at least one measurement. If multiple co-located measurements exist from a single mesh node, I consider the median of the measured SNR values. To determine the predictive accuracy of estimated mesh node's metric region, I weight the distribution of measurement distances, \( d(n, p) \), to give less weight to progressively longer distances. The weights are based on the empirically measured
distribution of distance from a client location to the nearest three mesh nodes.

The received signal power cannot be measured below a minimum receive power as the wireless card is not able to distinguish between weak transmissions and noise. In order to distinguish coverage holes from locations that the measurement study did not visit, I infer a coverage hole (uncovered location) with respect to a mesh node only if there is no measured value for that node but there is for another node. All measurements are obtained from the client, and hence all predictions pertain to the client performance.

Alternate Techniques

I compare the assessment framework against two alternate approaches: uniform propagation estimation and grid-based sampling. Uniform propagation restricts the shape of a metric region to a circle with radius determined only by the path loss exponent $\alpha$. In comparisons, I use the most accurate value of the radius for each circular region, assuming sufficient measurements have been taken. Second, grid-based sampling provides a fair comparison for a given number of measurements. For a given density, measurements are taken on a grid, and unknown points are estimated via interpolation.
5.4.2 Measurement Study Background

At the time of this study, the GoogleWiFi network consists of 447 Tropos mesh nodes mounted mostly on city light posts and covering a total outdoor area of 31 km² in Mountain View, California. The Tropos nodes consist of a 7.4 dBi antenna and a single 802.11g wireless interface. Figure 5.2 shows the digital map providing terrain feature information for GoogleWiFi, which is a publicly available economic zoning map of the city of Mountain View. I measured a 12 km² region, encompassing the northwest quadrant of the network, as shown in Figure 5.4. The client measurement platform for the GoogleWiFi study was a laptop with an external 802.11g wireless adapter, 3 dBi antenna, and GPS receiver.

The TFA Network is an urban mesh network, deployed in southeast Houston by Rice University [10]. At the time of the measurement study [11], the network consisted of 17 mesh nodes, providing coverage to a 3 km² area. Each mesh node features a high gain 15 dBi omnidirectional antenna placed approximatively 10 meters above the ground, higher than most of the houses and some of the trees in the neighborhood. The coverage region for the TFA network is shown in Figure 5.5. The TFA measurement platform was a laptop inside a car with an 802.11b wireless interface, a 7 dBi external antenna, and a GPS receiver.

The two networks have several key structural differences. The antennas used in TFA are taller and have higher gain, indicating a larger coverage region. This is offset
Figure 5.4  Coverage map in the GoogleWiFi network with circles representing covered locations and x's representing holes in the 12 km² measured area with 168 mesh nodes.

partially by the difference in terrain, as the TFA network is filled with larger, denser trees which act as attenuators. Moreover, the GoogleWiFi nodes are mounted on light poles along streets whereas most TFA nodes are mounted against houses in the interior of a residential block.

5.4.3 Coverage Metric

This section next focuses on validating the predictive accuracy of the framework for the coverage metric region based on SNR measurements from both GoogleWiFi and TFA. I separately investigate the terrain-informed estimation technique and boundary refinement algorithm. Also, I evaluate the degree to which the coverage metric obeys the monotonicity property and the resulting impact on framework
Figure 5.5 Coverage region of the TFA network with circles representing covered locations and x’s representing holes in 3 km² measured area, including the network edge where weak signal was measured.

For a given accuracy level, the framework reduces the required number of measurements by a factor of two to five as compared to a sampling and interpolation strategy. Figure 5.6 presents the predictive accuracy of my framework and a grid sampling strategy in both network scenarios as a function of the measurement budget per km². These values do not reflect a one-time overhead of approximately 60 training measurements. Note that both approaches have a practical upper limit to accuracy, regardless of measurement budget, which I later show is due to monotonicity violations. Moreover, grid sampling accuracy does not exceed the assessment framework’s accuracy
until approximately 10,000 measurements per km², i.e., an exhaustive study.

![Graph](image)

**Figure 5.6** Predictive accuracy of network coverage assessment with measurement budget, comparing the assessment framework with grid sampling strategy.

Figure 5.6 presents the total accuracy for covered and not covered locations. Though total accuracy improves, the accuracy for predicting covered locations is constant at 90% for all measurement budgets. As measurement budget increases, the ability of the framework to correctly predict coverage holes improves from 63% accuracy to 90% accuracy, leading to the improvements in Figure 5.6. Coverage holes are more difficult to correctly predict because all nearby mesh nodes' regions must be correctly predicted in order to correctly predict a hole.
Coverage Estimation Accuracy

I now focus on predicting a particular node's metric region, as opposed to the network's metric region by comparing four estimation strategies: 1) ideal estimation assuming all measurements are known, 2) per-location coverage estimation using Equation (5.1), 3) Algorithm Estimate-Mono-Metric-Region, and 4) the uniform propagation model. Note the difference between per-location estimation using only Equation (5.1) and estimation based on sectorization and boundary estimation is that the boundary-based estimation averages over a 20 degree sector before predicting locations.

*Terrain-informed estimation halves the estimation errors compared to uniform propagation.* Figure 5.7 presents the median accuracy of the four described estimation strategies. Terrain-informed estimation eliminates more than half of the errors resulting from the uniform propagation model. Surprisingly, the sectorization and boundary averaging has little effect in the GoogleWiFi network, but significantly impacts the TFA network. Increasing the number of sectors from 10 per node to 36 per node eliminates this difference in the TFA network. These results indicate that the TFA metric boundary locations are correlated within approximately 10 degrees due to larger variations in tree foliage than in the GoogleWiFi terrain. Note that grid sampling requires approximately 100 measurements per km² in order for interpolation accuracy to be 80%. For assessment on a per-node basis (e.g., for network
upgrades), this represents a relatively large cost increase as compared to estimation which requires no additional measurements.

![Graph showing predictive accuracy for the TFA and GoogleWiFi networks using uniform-propagation estimation, terrain-informed estimation, and ideal estimation.](image)

**Figure 5.7** Predictive accuracy for the TFA and GoogleWiFi networks using uniform-propagation estimation, terrain-informed estimation, and ideal estimation.

I now study the causes of errors in the estimation framework: monotonicity violations and the suboptimal choice of sector boundary. Monotonicity violations result in mispredictions even when the optimal sector boundary is chosen, which is a limitation of any framework that assumes a connected coverage region. The suboptimal boundary choice means that Equation (5.1) leads to a suboptimal choice of sector boundary location due to insufficient terrain feature information. For 31% of the measured sectors, the terrain-informed boundary estimation results in accuracy within 1% of the optimal boundary location, but for the remaining sectors, the terrain information
is not sufficient for perfect estimation.

*The probability of monotonicity violation is higher and has a stronger dependence on distance in the TFA network.* Figure 5.8 depicts the probability that a measurement farther from the mesh node has a better signal strength than a nearer measurement within a sector width of 0.1 degrees. The difference between the two networks is in part due to fact that the measured TFA network is mostly residential blocks without line-of-sight, whereas the GoogleWiFi environment features more open space and line-of-sight along streets (where nodes are mounted on light posts). When considering optimal accuracy in 10 degree sectors, 20% of the measured sectors in the GoogleWiFi network feature zero monotonicity violations and perfect accuracy, whereas this is true for only 1% of TFA sectors. Overall, non-monotonicity contributes 10-15% average error, but with a large range 0-40% per sector. The average error is only slightly (3%) higher for the TFA network, despite the greater violation probability. The range in sector accuracy is significantly smaller for TFA though, indicating that the GoogleWiFi results in Figure 5.8 are averaged over a broader variety of propagation environments. Note that I factor out temporal fluctuations by disregarding an SNR increase of less than 3 dB, as measurements show 90% of co-located measurements vary less than 3 dB.

*Requiring metric regions to be connected incurs a minimal penalty in accuracy.* In particular, adding a small number of disconnected metric regions has
Figure 5.8 Violations of the monotonicity property. The probability that signal strength increases by more than 3 dB when increasing radius in a sector of width 0.1 degrees.

a minor benefit (1.5%), even with ideally chosen boundaries. That is, if I allow a metric sector to also have one disconnected section, I then need to find three boundaries. Experiments show that even for boundaries that are found optimally, the mean accuracy increases from 89.8% to 91.4%. The remaining errors occur due to noisy metric boundaries.

Refinement with Measurements

I now study the refinement phase of the assessment framework using Algorithm Refine-Estimate. The experiments consider 10 degrees wide sectors with more than 50 measurement points spread throughout the sector. The simple ray-tracing algorithm
is also tested, with measurements taken within each sector to refine local values of $C_f$ terms.

The boundary refinement heuristic outperforms simple ray-tracing for improving accuracy with measurements. Figure 5.9 depicts accuracy as a function of the measurement budget and indicates that the refinement stage improves average estimation accuracy from 82% to 88%. The ideal estimation line indicates the accuracy with all measurements known and the optimally chosen sector boundary. Localized refinement of $C_f$ terms has small impact, indicating insufficient map granularity for simple ray-tracing techniques. The gain from the measurement refinement phase is less than the gain of exploiting terrain information for estimation, underscoring the importance of accurate $\alpha$ and $C_f$ parameters.

The framework's upper bound on the number of measurements is four times the actual needed number of measurements. There are two reasons Algorithm Refine-Estimate requires fewer measurements: boundary refinement stops early when a boundary is found and multiple mesh nodes are measured at a location with no extra cost. The first condition occurs because 30% of sectors require no more than three measurements to find the sector boundary. If the algorithm takes advantage of existing measurements within 15 meters of the estimated boundary instead of requiring a new measurement, the total required measurement budget is reduced by one-fourth. Note that I assume that taking an SNR measurement at point $p$ from
Figure 5.9 Accuracy of boundary refinement algorithm for the coverage region of mesh nodes in GoogleWiFi.

node \( n \) is approximately the same cost as taking an SNR measurement at \( p \) to all nodes \( n \in \mathcal{N} \).

Terrain parameter \( C_f \) estimation requires a moderate (10-20) number of measurements per terrain feature type. Equation (5.1) requires an estimated value of the average path loss exponent \( \alpha \) and the weight, \( C_f \), per terrain feature, to be determined as one-time overhead. I choose random measurements from the full set of measurements to estimate the average path loss \( \alpha \) and the \( C_f \) value of each terrain feature type. A moderately sized study of between 10 and 20 measurements achieves predictive accuracy within 2% of the best predictive accuracy. Note that there are
six feature types in the terrain map used for GoogleWiFi.

5.4.4 Modulation Rate Metric

The modulation rate metric captures the expected physical-layer modulation rate in use at a location, a value that depends on SNR and the rate selection protocol, e.g., ARF. This section uses the GoogleWiFi network to compare the boundary refinement algorithm in two scenarios: 1) measurement of the modulation rate directly and 2) refinement using coverage measurements and then mapping coverage to modulation rate boundaries. The second approach involves empirically mapping the modulation rate threshold $\theta_r$ to the coverage threshold $\theta_c$ which corresponds to the desired modulation rate region. I next describe in more detail these two approaches.

For the first refinement approach, the estimated modulation rate is calculated from the SNR measurements, using a piecewise linear function to map SNR value to expected modulation rate. This piecewise linear function maps the relationship for the specific network environment, though idealized values of this mapping can be obtained from the card manufacturer's specifications. The constant $C_r$ is a stepwise function with two cutoff SNR values, $C_1$ and $C_2$. Below $C_1$, the expected rate is the minimum (1 Mbps), and above $C_2$, the expected rate is the maximum (54 Mbps). Between $C_1$ and $C_2$, linear interpolation is used to find the expected rate. In the measurement study, I found $C_1 = 14$ dB and $C_2 = 32$ dB.

For the second refinement approach, the modulation rate measurements consist of
an SNR value and the current physical-layer modulation rate, sampled after transmission of periodic ICMP packets. These measurements reflect the extent of cross-traffic and hidden terminal effects in the GoogleWiFi network, which add noise to the linear mapping function. As expected, experiments show the modulation rate metric to be monotonic on average with respect to the coverage metric and therefore, to distance also.

![Graph showing comparison of boundary refinement strategies for modulation rate metric in GoogleWiFi, either directly measuring modulation rate values or estimating from coverage measurements.]

**Figure 5.10** Comparison of boundary refinement strategies for modulation rate metric in GoogleWiFi, either directly measuring modulation rate values or estimating from coverage measurements.

Directly measuring modulation rate regions is less accurate than first using measurements to refine coverage estimates. For metric boundary estimation, I compare two approaches: 1) directly measuring modulation rate for each sector and 2) measur-
ing SNR and then estimating modulation rate sector boundaries from refined coverage boundaries. Figure 5.10 evaluates the improvement due to measurement refinement on the mean predictive accuracy of the modulation rate region of a metric sector of width 10 degrees. Direct measurement is worse because the extra variation in modulation rate at a given SNR value adds noise and makes it harder to find the refined boundary.

Note that MAC and network-layer metrics require associating with the mesh node in question and small data transmissions, both of which significantly increase the required measurement time and further motivate using coverage measurements to refine boundary estimates. The one-time overhead of experimentally characterizing the mapping between SNR and modulation rate is approximately 20 random measurements to achieve within 1% of the best estimation accuracy.

5.4.5 Coverage Redundancy Metric

The estimation and refinement framework greatly increases accuracy at predicting 2-redundancy. I estimate the metric redundancy region using the estimated coverage metric regions, as it requires only coverage information. Figure 5.11 compares the distribution of the k-redundancy metric in the GoogleWiFi network with the estimated values from my framework and the uniform propagation model. For predicting if a location is 2-redundant, this translates to a uniform propagation accuracy of 50% and an accuracy of 84% for my framework. The requirement to correctly predict multiple
regions at each point accentuates the difference in accuracy between the framework and uniform propagation estimation. The relative accuracies in TFA (not shown) are approximately identical.

Figure 5.11  Distribution of the number of covering mesh nodes (redundancy) for each location in the GoogleWiFi network, comparing measured values with estimations.
5.5 Deployment Evaluation

Having validated the assessment framework, this section next applies the framework and measurement study to investigate the two deployed networks, TFA and GoogleWiFi. Specifically, I evaluate the effect of deployment density on coverage holes and study client association policies to understand the load-balancing qualities of deployments.

5.5.1 Coverage Holes and Deployment Efficiency

Here, I evaluate the efficiency of the two deployments in terms of the density of deployed nodes and their chosen locations. I first consider the sizes of the coverage holes in both networks, and then focus on the effect of deployment density on the likelihood of coverage holes.

*Half of measured locations without coverage are small holes, within 10 meters from a covered location.* I consider a measured location to be a coverage hole if it is not within 3 meters of a location in the coverage region of any mesh node. The size of a coverage hole is the distance to the nearest covered location. Contrasting estimation techniques, grid sampling with 30 measurements per km² predicts *three times* more coverage holes than exist, whereas the framework overpredicts by only 25%. This means that grid sampling approach would conclude that coverage holes are 3× more common and two to four times as large on average.

I next examine the distribution of deployment densities in each network with the
goal of understanding how to best deploy a network in terms of the coverage metric region. I compare two deployment strategies: minimizing the maximum distance to the nearest mesh node and the looser restriction of deploying at a specific localized node density. I reverse engineer both the TFA and GoogleWiFi networks to determine how to improve their deployment strategy.

I calculate the deployment density at a client location as local node density per km² within a circle around the client point with radius of 400 meters. Note that this is a client-centric definition of density, which allows us to evaluate locations of different density in a non-uniformly deployed mesh network. I focus on this specific localized region size because the data shows that less than 2% of client locations are covered only by a node farther than 400 meters away. Because of the different technology, antenna gains, heights, and propagation environment, the two networks are not expected to require the same node density. The mean density in TFA is 11.2 nodes per km², with standard deviation of 7.4. The mean density in GoogleWiFi is 17 nodes per km², with standard deviation of 5.9.

Deploying to minimize the distance from a client to the nearest mesh node leads to a 3× over-deployment in TFA. In Figure 5.12, I plot the probability of a coverage hole as a function of deployment density at a client point. If one seeks a deployment with 90% coverage, the maximum distance to a mesh node must be below 130 meters. This corresponds to a deployed density in a square grid of approximately 30 nodes
Figure 5.12  Probability of a coverage hole as a function of local deployment density for both networks.

per km\(^2\). However, Figure 5.12 indicates that only 11 nodes per km\(^2\) are required to attain 90% coverage. This difference is due to the fact that coverage is often provided by the mesh node that is *not* closest to the client. For a deployment in progress, the assessment framework can be used to determine the best density, per the above analysis.

In the GoogleWiFi network, the deployment density has little impact on coverage hole probability above 8 nodes per km\(^2\). Considering the nearest node deployment strategy, I find that in order to limit the probability of a coverage hole to less than 10% in GoogleWiFi, the distance to nearest node must be less than 80 meters, which
translates to a deployment density of approximately 77 mesh nodes per km². This density is dramatically higher than the actual deployed density of 17 nodes per km², although the actual probability of a coverage hole is approximately 25%. In order to achieve 90% coverage, client-side solutions, such as use of higher-gain antennas, may be more cost effective than the very dense deployment.

*Coverage holes in GoogleWiFi are correlated between mesh nodes, leading to significantly higher (4×) node density requirements to decreases holes from 25% to 10%.* Surprisingly, the TFA network's coverage-hole probability quickly approaches zero as density increases, whereas the GoogleWiFi network does not, indicating that further increasing node density has diminishing impact on coverage hole probability in GoogleWiFi. For all client locations, I examine each mesh node within 400 meters and the boundary distance for the sector that the location belongs to. For the coverage holes, I found that 75% of these sectors had boundary distance of at least 50 meters less than the average boundary distance of 178 meters. Therefore, the GoogleWiFi environments presents a significantly greater deployment challenge. *In general, improving the coverage of deployed mesh networks is challenging because the coverage holes are small and spread out, requiring many new node locations to remove them.*

### 5.5.2 Load-Balanced Node Deployment

I next investigate the load-balancing qualities of a deployment through investigating association policies for clients in range of multiple mesh nodes. The objective is to
show that given standard client association policies, there is significant imbalance in
the number of client locations each mesh nodes serves. I compare association based
on the client's strongest signal strength with an ideal policy that jointly considers
signal strength and load balancing.

To study this issue, I first define a method for calculating the load on each mesh
node under a hypothetical population-based model. In particular, I consider the
offered load of a location $i$ to be a function of the client demand $q_i$ (in kilobits) at $i$
and the expected time required to serve a fixed sized packet, $d_i$, (measured in seconds
per kilobit) at $i$. The framework provides the estimate for $d_i$ based on the predicted
modulation rate at location $i$, which is in turn based on the SNR at the location. Let
$Q_n$ represent the set of clients associated with mesh node $n$, then the load of node $n$
is $L(n, Q_n) = \sum_{i \in Q_n} d_i q_i$. As load is measured in time units, if $L > 1$, then the mesh
node is saturated. In this case, the clients' served load will be the offered load $q_i$
divided by $L$. Otherwise, the full client demand, $q_i$, is served. I further assume that
the access tier for each mesh node operates on an independent frequency channel,
and hence there is no interference between nodes.

I now describe two client association policies. Most existing clients employ a policy
in which they compare received SNR values of all APs within range and associate to
the node with the strongest SNR. For a load-balanced policy, I formulate and solve
the association problem as a maximum flow problem on bipartite graphs [51] with
one set of nodes as the client locations, the other set as mesh node locations, and an edge in the graph when a client node is in the coverage region of a mesh node. The supply is $d_i \times q_i$ for every client node $i$, the capacity of each edge emanating from $i$ is also $d_i \times q_i$, and the demand is 1 for every mesh node.

For the experiment setup, I vary the client demand from 100 kbps to 1 Mbps, and use the mapping from SNR to modulation rate presented in Section 5.4. The term $d_i$ is calculated as twice the inverse of the modulation rate to also account for overhead.

**Figure 5.13** Percentage gain in average client throughput from using a centralized, load-balancing association scheme versus realistic, local association with strongest SNR.

*Max-flow load-balanced association improves total client throughput by 20% in *GoogleWiFi* as compared to strongest-SNR association.* Figure 5.13 shows 15-20%
gain in average client throughput for the centralized, load-balancing policy versus local association. In a regular grid deployment, the local policy results in the same association as the load-balanced policy due to the fact that the same number and quality (SNR) of clients associate with each node. The loss in throughput of the strongest SNR policy is then also the loss due to the uneven deployment of mesh nodes, resulting in some overloaded nodes and some under-loaded nodes. Note that the most gain available is at moderate offered loads, where there is significant load imbalance on mesh nodes and some nodes have available capacity. The gain in the TFA network is less because a larger fraction of the client nodes have only one possible node to associate with.
5.6 Related Work

Measurement Strategies. While many previous studies present measurements of the coverage of a wireless network [52, 10, 11, 53, 8], none have proposed a framework for choosing the number and location of measurements to characterize a metric region via a small number of measurements. Ray-tracing performs detailed simulation and prediction of physical-layer propagation in order to estimate physical-layer propagation [53, 7, 8]. However, ray-tracing requires highly detailed information about the environment, such as building materials and thickness, to achieve high accuracy in outdoor environments. Other studies [52, 10] have used a small set of coverage measurements to estimate parameters such as path loss and shadowing, but assume a uniform propagation (circular) model. Finally, a recent work [12] studied a deployed urban mesh, including a small measurement study of client performance which confirmed that simple pathloss models fail to predict coverage holes, but they did not address the problem of characterizing coverage holes.

Physical Layer Models. The Okumura model [50] and related models [54] for outdoor propagation are widely-used empirical models to predict signal propagation in urban environments. However, they apply to different carrier frequencies, coarse-grained terrain features, and clients located more than 1 km from the base station. Terrain-informed estimation builds upon indoor propagation modeling techniques that use attenuation factors derived from building blueprints to show wall
locations, thickness, and material [55]. A good review of such propagation modeling techniques is in [15]. In contrast, the presented terrain-informed estimation is a sectorized technique applied to outdoor environments via the "push-pull" boundary refinement technique. Moreover, I operate with finer-grained resolution of terrain features and utilize training measurements to obtain localized estimates of attenuation.

**Cellular Coverage.** Cellular networks feature different frequency bands, antenna heights, and propagation environments, but still encounter related coverage assessment problems. Previous studies have addressed how to characterize coverage in a cellular network with a small number of signal strength measurements [56] and in the presence of random shadowing (violations of coverage monotonicity) [57, 58]. However, prior work uses uniform propagation models and does not provide strategies for selecting measurement locations. Likewise, "diamond-shaped" coverage regions were studied for Manhattan cellular networks [59], with exhaustive measurements used for a downtown-only environment.

**WLAN and Mesh Measurement Studies.** For indoor WLAN deployments, exhaustive measurements have been compared to propagation modeling tools [60]. Other measurement studies in mesh and WLAN networks focused on client usage and mobility [11, 61, 1] or protocol performance. A war-driving technique was proposed to study residential networks [62], but with a focus on measuring residential broadband speeds and not wireless performance. Finally, a related problem is node placement,
i.e., choosing locations to deploy wireless APs: this problem has been formulated as an optimization problem requiring as input exhaustive measurements [27]. In contrast, this framework can provide input to a placement algorithm with substantially smaller measurement overhead.
5.7 Summary

In this chapter, I present a measurement framework to accurately characterize the metric regions of a mesh node using only a small number of measurements. The primary metric I consider is coverage (signal strength), and the technique of estimating and refining metric regions can be extended to other monotonic metrics. I also use the framework to study modulation rate and redundancy metrics.

The presented framework utilizes publicly available terrain maps to improve coverage estimation, showing that coarse-grained maps significantly improve estimation accuracy. This improved estimation leads to fewer needed measurements to refine the estimated region boundaries. I further examine the sources of estimation error and find that coverage monotonicity violations account for an average of 10\% error, although with much greater variation per sector in GoogleWiFi than in TFA. With this framework, I then reverse-engineer the TFA and GoogleWiFi deployments and find that improving the performance of GoogleWiFi through the addition of nodes would be highly problematic (costly) due to the prevalence of numerous coverage holes having small area.
Chapter 6
Conclusions

This thesis focuses on designing algorithms for the deployment and assessment of wireless mesh networks in enable lower-cost mesh networks.

6.1 Thesis Contributions

Chapter 3 presents algorithms for the mesh node placement (MNP) problem. We first formulate the MNP 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 and connectivity constraints, through the construction of a Terminal Steiner tree on a discrete input graph consisting of both coverage locations and potential mesh node locations. As a result, our algorithm has the best-known approximation ratio (3.1) for the problem of minimizing the average contention at a client location. We also present a second algorithm that minimizes deployed nodes using first the estimated signal strength values in the input graph and then iteratively measuring a small number of backhaul links in the solution TST. As our result, this algorithm ensures that the backhaul-tier is fully connected in the final deployment, while also requiring $1000\times$ fewer measurements than a complete measurement study.
We then evaluate the performance of our algorithms, showing an 80% improvement respectively in measurement-based non-uniform propagation scenarios.

Chapter 4 studies the gateway placement problem, first introducing a technique to efficiently compute gateway-limited fair mesh capacity as a function of the contention at each gateway. We then present two gateway placement algorithms adapted from local search heuristics for related facility location problems with provable performance guarantees. The MinHopCount algorithm adapts a local search algorithm for the capacitated facility location problem and utilizes minimum wireless hop count local operations to maximize capacity, iteratively estimating the gateways' wireless capacities. The MinContention algorithm is adapted from a solution to the uncapacitated $k$-median problem and minimizes the average contention region size within a provable approximation ratio of $3 + \varepsilon$. MinHopCount is more general and can handle non-uniform gateway costs, while MinContention is able to provide better performance guarantees. Our numerical results on three real topologies show that our algorithms outperform a greedy heuristic and achieve close to the optimal capacity. Further, we show that near-optimal solutions have similar gateway configuration as the optimal, but the difference in capacity is large, which supports the use of local search operations on near-optimal placements.

Chapter 5 presents a measurement framework to accurately characterize the metric regions of a mesh node using only a small number of measurements. The primary
metric we consider is coverage (signal strength), and the technique of estimating and refining metric regions extends to other monotonic metrics. We utilize publicly available terrain maps to improve coverage estimation, showing that coarse-grained maps significantly improve estimation accuracy. This improved estimation leads to fewer needed measurements to refine the estimated region boundaries. We further examine the sources of estimation error and find that coverage monotonicity violations account for an average of 10% error, although with much greater variation per sector in GoogleWiFi than in TFA. With the assessment framework, we then reverse-engineer the TFA and GoogleWiFi deployments and find that improving the performance of GoogleWiFi through the addition of nodes would be highly problematic (costly) due to the prevalence of numerous coverage holes having small area.

6.2 Future Work

The research done in this thesis points to several directions for future study. First, the deployment problem for mesh networks has many other aspects not yet considered, such as the number of radio interfaces in each mesh node and the channel assigned to each. The channel assignment problem has been studied extensively (e.g., cite), but only for a set number of radio interfaces. I have previously shown the benefit of adding an additional radio to separately handle all access traffic [42], but did not examine the impact of adding radios at only key nodes, e.g., capacity points. Additionally, there are many deployment parameters, such as transmit power, not yet considered,
nor has a WiMax-based capacity tier been explicitly considered. This thesis opens many doors to deployment-related objectives, where prior research normally assumes a fixed topology and then seeks to optimize over that, e.g., scheduling or routing. My research suggests that protocol-aware placement of mesh resources will lead to higher performance and even enable new techniques only possible because of the purposefully designed topology.

A related problem to mesh network deployment is the continuous monitoring of an operational mesh network. Detecting network faults or anomalies is difficult because 1) event detection is distributed in nature and 2) the bandwidth available for monitoring information is highly constrained. In order to differentiate between a network error or malfunction, a link going into deep fade, or an malicious attack, the perspectives from multiple mesh nodes must be compared, as each sees a different set of signals. Exchanging information between mesh nodes or to a centralized location is challenging because that data must use the same, already-constrained bandwidth as the network clients. The constraints suggest a selective sampling and coordination protocol, where the objective is to detect all anomalies at a given accuracy and delay while exchanging minimal control messages.
References


