Capacity Characterization of Multi-Hop Wireless Networks- A Cross Layer Approach

Deepti Ramesh Chafekar

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Dr. Anil Vullikanti, co-chair
Dr. Madhav V. Marathe, co-chair
Dr. Godmar V. Back
Dr. Christopher L. Barrett
Dr. Y. Thomas Hou
Dr. Aravind Srinivasan

March 11, 2009
Blacksburg, Virginia

Keywords: Wireless networks, cross-layer design, interference, SINR model, approximation algorithms

Capacity Characterization of Multi-Hop Wireless Networks- A Cross Layer Approach

Deepti Ramesh Chafekar

Abstract

A fundamental problem in multi-hop wireless networks is to estimate their throughout capacity. The problem can be informally stated as follows: given a multi-hop wireless network and a set of source destination pairs, determine the maximum rate r at which data can be transmitted between each source destination pair. Estimating the capacity of a multi-hop wireless network is practically useful — it yields insights into the fundamental performance limits of the wireless network and at the same time aids the development of protocols that can utilize the network close to this limit. A goal of this dissertation is to develop rigorous mathematical foundations to compute the capacity of any given multi-hop wireless network with known source-destination pairs.

An important factor that affects the capacity of multi-hop wireless networks is radio interference. As a result, researchers have proposed increasingly realistic interference models that aim to capture the physical characteristics of radio signals. Some of the commonly used simple models that capture radio interference are based on geometric disk-graphs. The simplicity of these models facilitate the development of provable and often conceptually simple methods for estimating the capacity of wireless networks. A potential weakness of this class of models is that they oversimplify the physical process by assuming that the signal ends abruptly at the boundary of a geometric region (a disk for omni-directional antennas). A more sophisticated interference model is the *physical interference* model, also known as the *Signal to Interference Plus Noise Ratio* (SINR) model. This model is more realistic than disk-graph models as it captures the effects of signal fading and ambient noise. This work considers both disk-graph and SINR interference models.

In addition to radio interference, the throughput capacity of a multi-hop wireless network also depends on other factors, including the specific paths selected to route the packets between the source destination pairs (routing), the time at which packets are transmitted (scheduling), the power with which nodes transmit (power control) and the rate at which packets are injected (rate control). In this dissertation, we consider three different problems related to estimating network capacity. We propose an algorithmic approach for solving these problems. We first consider the problem of maximizing throughput with the SINR interference model by jointly considering the effects of routing and scheduling constraints. Second, we consider the problem of maximizing throughput by performing adaptive power control, scheduling and routing for disk-graph interference models. Finally, we examine the problem of minimizing end-to-end latency by performing joint routing, scheduling and power control using the SINR interference model. Recent results have shown that traditional layered networking principles lead to inefficient utilization of resources in multi-hop wireless networks. Motivated by these observations, recent papers have begun investigating cross-layer design approaches. Although our work does not develop new cross-layered protocols, it yields new insights that could contribute to the development of such protocols in the future.

Our approach for solving these multi-objective optimization problems is based on combining mathematical programming with randomized rounding to obtain polynomial time approximation algorithms with provable worst case performance ratios. For the problems considered in this work, our results provide the best analytical performance guarantees currently known in the literature. We complement our rigorous theoretical and algorithmic analysis with simulation-based experimental analysis. Our experimental results help us understand the limitations of our approach and assist in identifying certain parameters for improving the performance of our techniques.

Acknowledgments

I wish to thank my advisors Dr. Anil Vullikanti and Dr. Madhav Marathe for their constant guidance and encouragement. They have mentored me through different aspects of the Ph.D. program. This includes reading and writing papers, approaching research problems, making presentations and giving talks. They have played a vital role in shaping my attitude towards research and work. I am especially indebted to Dr. Anil for working with me on a day-to-day basis and being patient with me. He has constantly motivated me and boosted my morale when my spirits were down. I am grateful to be associated with my advisors and to have had the opportunity to work with them. Their positive attitude, attention to detail, pursuit of excellence and perfection, passion towards research, commitment and sincerity towards work has inspired me all through my Doctoral program. I hope to imbibe these qualities as I step into my professional career.

I am grateful for the assistance of my committee members: Dr. Godmar Back, Dr. Aravind Srinivasan, Dr. Thomas Hou, and Dr. Chris Barrett. Each of them have been supportive of my efforts and have provided constructive feedback on my work on a timely basis.

I owe a great deal to the entire staff and faculty of NDSSL (Network Dynamics and Simulation Science Labs). They have made my work environment pleasurable. I would especially like to thank Joyce Randall for her motherly warmth and comfort. I am thankful to my colleagues and fellow students at NDSSL for their friendly disposition. They include Bryan Lewis, Karthik Channakeshava, Chris Kuhlman, Tridib Dutta, Elaine Nsoesie, Guanhong Pei, Jonathan Leidig, Fei Huang, Sharon Smyth, and Ginger Hansen. My special thanks to

Dr. Achla Marathe and Aparna Vullikanti for their warm hospitality.

I would like to thank my good friends and well-wishers Andrea Apolloni and Karthik Channakeshava for always standing by my side. I have had several stimulating discussions with Karthik, and he has assisted me on several occasions with his expertise on NS-2. Andrea is a fountain of joy and energy. He has always managed to make me smile during stressful times.

I owe special thanks to Dr. S.S. Ravi (University at Albany-State University of New York) for his interest in my work and professional development. His modest disposition and deep sincerity towards work has always inspired me.

All the results in this dissertation were obtained in collaboration with my advisors, Dr. Anil Vullikanti, and Dr. Madhav Marathe. In addition, the results in Chapters 4 and 7 are due to joint work with Dr. Srinivasan Parthasarathy (IBM T.J. Watson) and Dr. Aravind Srinivasan (University of Maryland). Dr. Srinivasan Parthasarathy has also collaborated with me for the results in Chapter 6. I would also like to thank my other collaborators Dave Levin, Karthik Channakeshava and Dr. Keith Bisset.

Finally, I would like to thank my family - my husband Shrirang Yardi, my parents, my sister Preeti and my in-laws for their unconditional love. Thanks for always believing in my abilities and supporting me through thick and thin. I am eternally grateful to Shrirang for encouraging me to pursue my Doctoral program. I would like to thank him for his immense patience, kindness, and understanding. I am grateful to my parents for the countless sacrifices they have made for my sake.

Dedication

To Mummy, Papa and Shrirang

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Wireless networks are widespread and are finding an increasing number of applications in our day-to-day lives. Despite significant engineering and practical advances in wireless networking technologies, there is limited understanding of the fundamental limits of their performance and efficiency. One of the most important problems in wireless networks is to estimate their throughput capacity. This problem can be informally stated as: given a set of wireless transceivers and a set of source-destination pairs, *determine the maximum data rate that can be achieved over all the connections.* The knowledge of network capacity can (a) provide insights into performance limits of existing protocols, (b) assist in the development of techniques that can utilize the network close to its capacity, and (c) aid in effective planning and deployment of wireless networks. Starting from the classical work of Shannon [15], there has been significant research on characterizing the capacity of wired networks. However, a similar characterization for wireless networks remains an open research problem. *The goal of this dissertation is to develop rigorous mathematical foundations to compute the capacity of any given multi-hop wireless network with known source-destination pairs.*

This work focuses on multi-hop ad-hoc wireless networks which are formed by a group of communicating wireless devices that operate without any fixed infrastructure (cf. Figure 1.1). These networks are self-configurable and have low set-up costs. Due to these features, they are a popular choice in scenarios such as habitat monitoring, providing communication in military and disaster zones and enabling "smart homes" (where computers and various appliances such as microwave ovens, refrigerators etc. are interconnected by a wireless network). However, since these networks are not supported by any infrastructure, they also suffer from some limitations such as (a) limited capacity and (b) rigid constraints on power utilization. Hence, it is important for these networks to use the available capacity efficiently while conserving as much power as possible.

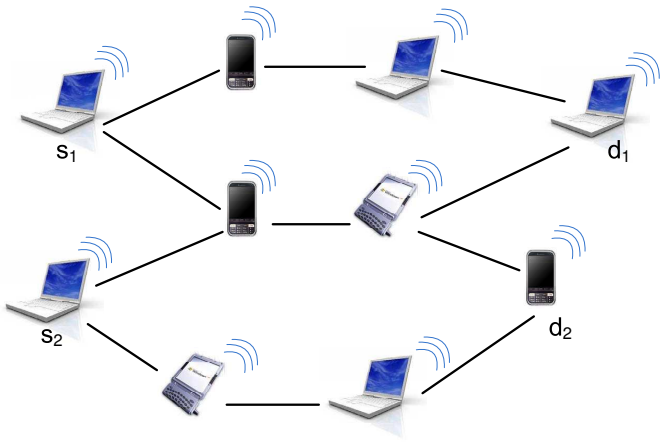


Figure 1.1: Example of a multi-hop wireless network. Two source-destination pairs are represented as s_1, d_1 and s_2, d_2. An edge exists between two devices if they lie in each other's wireless transmission range. Routes are constructed dynamically in which devices cooperate by forwarding each other's data packets.

The problem of capacity characterization of multi-hop wireless networks has been extensively researched. An influential result in this area is that of Gupta and Kumar [21]. They show that in a random network, where n identical transceivers (nodes) are distributed uniformly

in a unit square and each node communicates with a random destination, the uniform per node throughput capacity in bit-meters/second is $\theta(\frac{1}{\sqrt{n \log n}})$. This result indicates that as the number of nodes in the system increases, the per-node capacity scales down significantly. The result by Gupta and Kumar is a scaling law for random networks and has limited relevance to practical networks. Recent studies by Jain *et al.* [22], Kodialam *et al.* [25, 26], Kumar *et al.* [30], Alicherry *et al.* [3] and Lin *et al.* [35] have investigated the capacity of *any* given multi-hop wireless network. These studies propose mathematical techniques for deriving upper and lower bounds on network capacity. Our work has been motivated by these studies. However, a key distinction of our work is that we consider a more complex and realistic wireless interference model and provide mathematical techniques using this model to derive bounds on network capacity. In the next subsection, we explain the wireless interference models in detail and discuss some of the challenges we face in this work.

1.2 Challenges

The capacity of a multi-hop wireless network depends on various factors such as: (a) protocols at different layers of the Open Systems Interconnection (OSI) model [59], (b) wireless interference and models (such as disk-graph based and SINR based) used to capture it, (c) fairness desired among different connections, and (d) constraints on the total power consumed and total tolerable delay. There is significant interplay between these factors, which makes it extremely challenging to incorporate them in a single formulation [21, 22]. In this dissertation, we study the capacity estimation problem by considering a combination of these factors. Capacity also depends on the mobility of transceivers, but we do not consider mobility effects in this work. In this section, we elaborate the main challenges we faced and the techniques we used to address them.

1.2.1 Cross-layer Interaction

Existing network protocols utilize the layered architecture derived from the OSI model. Informally, in this model, communication functions are distributed across different layers such as physical, medium access control (MAC), network, transport and application. Recent research has suggested that this model can lead to inefficient utilization of resources and capacity benefits are possible by inducing interaction between these layers [4, 37]. For example, in a study conducted by Barrett *et al.* [4], the authors empirically demonstrate that the routing protocol can significantly affect the MAC layer protocol and vice-versa. The authors conclude that performance of the MAC and routing layers cannot be accurately characterized by considering them in isolation. A cross-layer design extends the traditional OSI layered architecture by alteration of interfaces to enable interaction and information exchange between different layers of the protocol stack [12, 37]. However, this feedback mechanism imposes additional overheads in cross-layer design and increases its modeling complexity.

In our work, we use *mathematical programs* to capture cross-layer constraints. Due to complexity of the cross-layer constraints, these programs are either integer linear or non-convex. We therefore develop *provable* approximation algorithms that (a) transform these complex programs into linear programs, (b) run in *polynomial time*, and (c) provide *guarantees* on the quality of the solution with respect to the optimal solution.

1.2.2 Wireless Interference

An important factor that affects network capacity is *wireless interference*. Due to the broadcast nature of the wireless medium, transmissions from one sender interfere with the transmission and reception capabilities of other nearby nodes [46]. Wireless interference is complex and difficult to model. A commonly used class of models for wireless interference is that of a *disk-graph* [6, 28, 50, 51]. In these models, a disk of radius $r(u)$ is associated with every

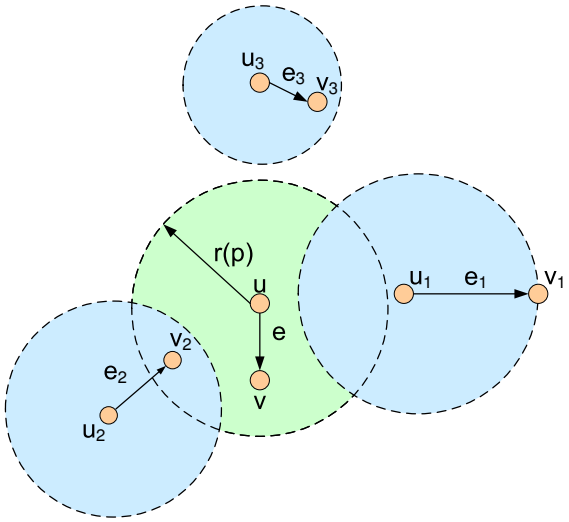


Figure 1.2: Tx-Rx interference model. A disk is associated with every sender. Links $e = (u, v)$ and $e_2 = (u_2, v_2)$ interfere with each other as receiver v_2 lies in the disk associated with node u.

transceiver (node) u. This radius is proportional to the transmission power of the node. A link $e = (u, v)$ exists between sender u and receiver v, if the Euclidean distance $d(u, v) \leq r(u)$. Informally, links within "close" vicinity of each other are said to interfere with each other. Figure 1.2 shows an example of one such disk-graph based model known as the Tx-Rx model. In this model, any two links interfere with each other, if their senders or receivers lie in each other's disk. These models provide a useful first approximation for understanding wireless interference; they localize the interference, which in turn, facilitates the development of simple and provable methods for estimating the capacity of wireless networks. For example, using these models, the link scheduling problem (which determines when each link should transmit) can be reduced to a well studied graph coloring problem and can be solved by greedy heuristics. However, these models oversimplify the actual interference by assuming that the signal from a radio can only be heard within a disk and that signal collisions always lead to lost messages.

A more sophisticated interference model is the *physical interference* model, also known as the *Signal to Interference Plus Noise Ratio* (SINR) model [21, 40, 42]. In this model, a signal from a transmitter u is said to be successfully received by a receiver v, if the ratio of u's signal strength at v and the combined interference from other transmitters, along with ambient noise, exceeds v's antenna gain. The SINR model is believed to be more realistic than the disk-graph models as it captures the effects of signal fading and ambient noise [41]. In this dissertation we study both disk-graph and SINR interference models.

1.3 Contributions

The central contribution of our work is an algorithmic framework for approximating network capacity of any given multi-hop wireless network. Specifically, our contributions are:

1. We study three aspects of the capacity estimation problem: (a) approximating throughput capacity, (b) understanding the impact of adaptive power control on overall throughput capacity, and (c) minimizing overall latency. We consider the impact of cross-layer constraints when modeling each aspect.

2. We combine two powerful techniques - *linear programming* and *randomized algorithms* - to derive novel polynomial time approximation algorithms with provable worst case approximation ratios. We develop new geometric packing analyses to derive the upper and lower bounds on the throughput capacity. For the problems considered in this work, our results provide the best analytical performance guarantees currently known in the literature.

In the following subsection, we provide more details on each of the problems studied and our results.

1.3.1 Problems Studied

We model the wireless network as a graph $G(V, E)$, where V denotes the set of transceivers (referred to as nodes) located in the Euclidean plane and $\mathcal{E} \subseteq V \times V$ denotes the set of possible directed links (referred to as edges) on which transmissions can occur. Let $\mathcal{K} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ denote the set of k -destination pairs and $\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ denote the set of packets for every source Table 1.1 summarizes the problems considered in this work.

Problem	TM-SINR	PETM	CLM
Objective	Max throughput	Max throughput	Min Delay
Components	Routing, Scheduling, Rate assignment	Routing, Scheduling, Rate assignment, Adaptive power control	Routing, Scheduling, Power control
Interference Model	SINR	Disk-graph	SINR
Approximation factor	poly-logarithmic	constant	poly-logarithmic

Table 1.1: Problems studied in this work.

Problem 1: Throughput Maximization with SINR constraints (TM-SINR)

We first consider the problem of maximizing overall throughput of any given multi-hop wireless network. We model interference using the SINR model. For given values of V, $\mathcal{K}$, and a power level vector $\bar{J}_{edge}$ that specifies power level $J(e)$ for transmission on each link e, the objective of TM-SINR problem is to maximize total throughput capacity by: (i) choosing routes for the connections, (ii) choosing data rates on the routes, and (iii) scheduling the packets at each time slot, such that SINR constraints are satisfied for all simultaneous transmissions.

Results and Contributions:

- We construct network instances and compare the throughput predicted by the SINR and disk-graph model for these instances. We observe that the predicted throughput capacity is significantly different in these two models. For the case of uniform power levels (the case when power level for all edges is the same), we show that there are instances in which throughput capacity predicted by the SINR model is significantly higher than that predicted by the disk-graph model. For the case of linear power levels

(the case when power level for each edge is $J(e) \propto \ell(e)^{\alpha}$, where $\ell(e)$ denotes the edge length), we show that there are instances in which throughput capacity estimated by the disk-graph model is higher than that estimated by the SINR model. Our results suggest that both interference models have their own limitations and greater care needs to be taken while using these models.

- We use linear programming techniques to derive upper and lower bounds on the overall throughput capacity. We develop a polynomial time approximation algorithm that provides a feasible rate vector whose total throughput is at most a *poly-logarithmic* factor away from the optimal throughput. This approximation factor is a worst case guarantee that holds for every instance of the TM-SINR problem.

- For two special cases of power level choices - uniform and linear power levels - we improve the poly-logarithmic approximation factor to a logarithmic factor.

- We develop a geometric packing insight for characterizing and reasoning about interference in wireless networks with the SINR interference model and introduce a new congestion measure that incorporates SINR constraints. The upper and lower bounds derived for the TM-SINR problem depend on this measure. We also develop a routing metric using this congestion measure and experimentally show that the routes selected using this metric can yield a higher throughput than hop-count based shortest path routes.

- We conduct simulations to validate our theoretical model and demonstrate that the data rates obtained by our model are achievable in a realistic setting. We compare the results of our approximation algorithm with the optimal solution for small network instances. We observe that the upper and lower bounds derived by our technique are worst-case bounds and the approximation guarantee achieved in practice is much better than that predicted by our model. These experimental results help us understand the limitations of our approach and assist in identifying certain parameters for improving the performance of our model.

Problem 2: Power efficient throughput maximization (PETM)

We study the problem of joint optimization of total throughput rate and total power used, in which nodes have the capability to adaptively choose power levels. For given values of V, $\mathcal{K}$, a set $\mathcal{J}$ of power level choices and a limit B on the total power consumed, the objective of the PETM problem is to maximize the total throughput rate and ensure that the total power used is at most B, by choosing: (i) routes for each connection, (ii) an interference free schedule for all the links, (iii) power level for each link at each time slot, and (iv) data rates for each connection. For this problem, we model interference using disk-graph models.

Results and Contributions:

- The PETM problem can be naturally modeled as a non-convex optimization problem. We develop techniques for relaxing this non-convex program into a linear program (LP) and then use this LP to derive upper and lower bounds on the overall throughput capacity. We develop a polynomial time approximation algorithm that provides a feasible rate vector whose total throughput is at most a *constant* factor away from the optimal throughput. Our LP formulation is generic and can accommodate additional constraints and different objective functions.

- We conduct simulations to compute explicit tradeoffs between various quantities such as total throughput, total power used, number of connections, and long term fairness.

Problem 3: Cross Layer Latency Minimization with SINR constraints (CLM)

We study the problem of minimizing latency (length of the end-to-end schedule) in wireless multi-hop networks. We model interference using the SINR model. For given values of V, $\mathcal{K}$, $\mathcal{Q}$ and the minimum and maximum power levels j_{min}, j_{max}, respectively, the objective of the CLM problem is to minimize overall latency by: (i) assigning power levels to individual transceivers, (ii) choosing routes for the packets, and (iii) constructing an end-to-end schedule for the packets, such that all the transmissions that happen simultaneously satisfy the SINR interference constraints.

Results and Contributions:

- We develop a polynomial-time randomized approximation algorithm for the CLM problem that guarantees overall latency to be at most a *poly-logarithmic* factor away from the optimal latency.

- As corollaries of the CLM problem, we obtain polynomial-time randomized approximation algorithms for the following additional problems: (a) joint routing and end-to-end scheduling to minimize latency when the power levels are pre-specified for all nodes; (b) end-to-end scheduling to minimize latency when both routes and power levels are pre-specified; and (c) the CLM problem with an additional constraint that the total energy consumed is at most some given total power B.

1.4 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, we introduce concepts such as cross-layer protocols, approximation algorithms and the role of linear programming and probabilistic methods in the design of approximation algorithms. A basic understanding of these concepts is required to comprehend the results discussed in this work. In this Chapter, we also review the basic models, definitions and notations used. In Chapter 3, we provide an extensive review of prior work in the area of capacity characterization. Chapter 4 discusses our methodology for solving the TM-SINR problem. We present the experimental results for the TM-SINR problem in Chapter 5. Chapters 6 and 7 discuss solutions for the PETM and CLM problems, respectively. Finally, we present our conclusions and the main open problems inspired by our work in Chapter 8.

Chapter 2

Preliminaries and Model

In this chapter, we review the basic models, definitions and notations used in this work. We first provide a brief primer on cross-layer protocols and approximation algorithms. Linear programming based techniques and randomized methods play a vital role in the design of the approximation algorithms proposed in this dissertation. We provide an overview of these techniques. Table 2.1 gives a complete list of all notation used in this work.

2.1 Cross-layer Protocols

Existing network protocols use the layered structure derived from the Open Systems Interconnection (OSI) model. Informally, in this model, the communication functions are broken down into different layers such as physical, MAC, network, transport and application. Each layer is responsible for a fixed set of tasks and only has to concern itself with the interfaces with its adjacent layers. The advantage of this architecture is that it achieves modularity so that changes in layers can be made independently. While this architecture has been known to be effective for wired networks, it may not be adequate for wireless networks [37]. There are certain unique challenges in wireless networks that do not allow a direct application

of techniques and models from the wired setting. In particular, the wireless medium is multi-access and simultaneous transmissions interfere with each other. The channel capacity depends on this interference and on the surrounding ambient noise [1]. This causes interdependencies across the network layers that are not present in wired networks. The cross-layer approach suggests that by unifying wireless resources across multiple layers of the protocol stack, the overall network operation can be optimized in a joint manner. For example, in a study conducted by Barrett *et al.* [4], the authors empirically demonstrate that the routing protocol can significantly affect the MAC layer protocol and vice-versa. The authors conclude that performance of the MAC and routing layers cannot be accurately characterized by considering them in isolation.

Although cross-layer designs offer a great deal of flexibility, they are typically complex to solve [37]. For instance, interference causes fairly high scheduling complexity in wireless networks. The scheduling mechanism needs to carefully select a subset of links that can transmit at the same time. Further, the capacity of each link depends on the power and transmission schedule of other links. This relationship between the link capacity, power assignment, and the transmission schedule is typically non-convex. Therefore, the scheduling problem itself is a difficult non-convex problem to solve. Different approximation schemes and heuristics have been developed for designing cross-layer protocols. We refer interested readers to the work by Lin *et al.* [37] and Chen *et al.* [12] that provides an extensive survey of various cross-layer techniques.

2.2 Approximation Algorithms

As mentioned in the previous section, cross-layer design poses various optimization problems that are non-convex in nature. We therefore focus on developing approximation algorithms. An approximation algorithm for an optimization problem is an algorithm that produces in polynomial time, a feasible solution whose objective function value is "close" to the optimal.

The term "close" refers to a guaranteed factor of the optimal [57]. This factor is known as the *approximation ratio* or the *approximation guarantee* and can be used to measure the quality of the approximation algorithm. We now formally define the notion of an approximation algorithm.

Given an optimization problem $\mathcal{P}$ and an instance $\mathcal{I}$ of $\mathcal{P}$, let $OPT(\mathcal{I})$ denote the optimal objective-function value for $\mathcal{I}$; each feasible solution for $\mathcal{I}$ will have a non-negative objective-function value. If $\mathcal{P}$ is a maximization problem, an approximation algorithm $\mathcal{A}$ for $\mathcal{P}$ is an efficient algorithm that, for some $\gamma \geq 1$, produces a feasible solution of value at least $\frac{OPT(\mathcal{I})}{\gamma}$ for all instances $\mathcal{I}$ of $\mathcal{P}$. $\mathcal{A}$ is called a γ-approximation algorithm for $\mathcal{P}$. γ is the approximation guarantee or approximation bound for algorithm $\mathcal{A}$. Analogously, if $\mathcal{P}$ is a minimization problem, an approximation algorithm $\mathcal{A}$ for $\mathcal{P}$ is an efficient algorithm that, for some $\gamma \geq 1$, produces a feasible solution of value at least $\gamma \cdot OPT(\mathcal{I})$ for all instances $\mathcal{I}$ of $\mathcal{P}$ [53].

In order to derive a suitable approximation ratio for an approximation algorithm, one needs to compare the objective function value of the optimal solution with that of the approximate solution. However, for many cases, finding the optimal solution is itself a NP-complete problem. For such events, one needs to derive a "good" polynomial time computable *upper bound* for a maximization problem (or analogously a *lower bound* for a minimization problem) on the function value of the optimal solution. The approximate solution can then be compared with this upper bound (or lower bound as appropriate). Deriving these bounds typically requires unraveling the relevant structures that lie within these problems. Various algorithmic techniques have evolved over the past few years that can explore these inherent structures [57]. The most effective techniques in literature include greedy algorithms, mathematical programming based techniques, primal dual schemes, randomized techniques, and local search heuristics. In our work, we extensively use mathematical programming based techniques (mostly linear programming) and randomized algorithms for deriving the lower and upper bounds for various optimization problems. We now briefly review these techniques in the subsequent sections.

2.3 Linear Programming in Approximation Algorithms

Linear programming is the problem of optimizing a linear function (objective function) subject to linear inequality constraints. These linear programs are well understood and numerous efficient techniques have been developed to solve them exactly [57]. In this work, we use linear programming techniques to incorporate cross-layer constraints and approximate the original NP-complete optimization problem. Given an instance $\mathcal{I}$ of an optimization problem $\mathcal{P}$, we obtain a *linear relaxation* of $\mathcal{I}$ using a linear program (LP), where linear relaxation refers to the process of converting non-linear constraints into linear constraints. We demonstrate this in the following edge disjoint paths problem example [24].

Consider: (i) a graph $G(V, E)$, where V denotes the set of nodes and E the set of edges; (ii) a set of source-destination pairs $\mathcal{K} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$; (iii) a set of possible edge disjoint paths $\mathcal{P}_i$ for each source-destination pair i and let $B = \{P_1, \ldots, P_k\}$. Let $C(e) = |\{\mathcal{P}_i \in B : e \in \mathcal{P}_i\}|$ denote the number of paths an edge e belongs to. We define congestion $C = \max_{e \in E}\{C(e)\}$ and let $A(e) = \cup_i \{P \in \mathcal{P}_i : e \in P\}$ denote the set of paths that pass through e. We consider the problem of selecting a single path from the set $\mathcal{P}_i$ for each source-destination pair i, such that overall congestion C is minimized. This problem can be formulated as an *integer program* as follows[1]:

$$\min C \qquad \text{subject to:}$$

$$\forall i \in \{1, \ldots, k\}, \sum_{P \in \mathcal{P}_i} x(P) = 1 \tag{2.1}$$

$$\forall e \in E, \sum_{P \in A(e)} x(P) \leq C \tag{2.2}$$

$$\forall i \in \{1, \ldots, k\}, \forall P \in \mathcal{P}_i,\ x(P) \in \{0, 1\} \tag{2.3}$$

The variable $x(P)$ is an indicator for the path P being chosen. The constraints (2.1) ensure that exactly one path is selected for each connection i and the constraints (2.2) imply that

[1] We consider an elaborate version of this problem in Chapter 7

the maximum congestion on any edge is at most C. This integer program is known to be NP-hard even for simple graph classes (i.e. no efficient algorithm running in polynomial time is known that can solve this program) [24]. However, we can obtain a linear relaxation by replacing the integrality constraints 2.3 with linear constraints of the form

$$\forall i \in \{1, \ldots, k\}, \forall P \in P_i,\ x(P) \geq 0.$$

This linear relaxation transforms the integer program to a solvable linear program and *enlarges* the solution space of the problem in such a way that the objective function can be efficiently optimized over this enlarged set. The only difficulty with this scheme is that, owing to the increased solution space, the LP solution may not always yield a feasible solution to the original instance of the problem. This implies that the LP solution may result in selecting multiple paths for a given source-destination pair. A scheme, termed as *rounding*, is used to convert the LP solution to a feasible solution. It signifies the fact that the LP is obtained by relaxing integrality constraints, and the rounding algorithm needs to convert the potentially fractional values in the solution to appropriate integers [44]. If one can ensure that the solution obtained by the rounding scheme is feasible and is at most a factor of γ away from the optimal solution, a $\gamma-$ approximation algorithm can be achieved for the original optimization problem.

The key to developing a good approximation algorithm lies in developing efficient rounding techniques that can achieve an approximation ratio that is close to 1. In this dissertation, we explore two such rounding techniques - greedy algorithms, and *randomized* rounding (explained in the next section). Thus by combining LP relaxation techniques with appropriate rounding techniques, one can achieve an approximation algorithm for an optimization problem.

2.4 Randomized Algorithms

Randomized algorithms are algorithms that make random choices during their execution (Mitzenmacher and Upfal [39]). The forte of these algorithms is that the inherent randomness leaves a very small possibility for these algorithms to perform poorly for any given input. Moreover the performance of these algorithms is measured in *expectation* (average over possible random choices) or *with high probability (w.h.p)* on any given input. In most cases, randomized algorithms are simpler to design and are faster than their deterministic counterparts. However, the solutions obtained by these algorithms have some probability of being incorrect. The key in designing efficient randomized algorithms is to ensure that this probability of error is sufficiently small. We briefly discuss two specific ways in which randomization is employed, both in the context of this dissertation, as well as broadly in the design of algorithms.

Recall from Section 2.3 that for any given instance $\mathcal{I}$ of an optimization problem $\mathcal{P}$, linear programming approximation algorithms based on LP rounding consist of two steps: (i) linear relaxation and, (ii) rounding. Randomized rounding refers to the use of randomization to map an optimal solution computed by the LP to a solution that is indeed feasible for $\mathcal{I}$ [53]. For instance, let us consider the integer program discussed in Section 2.3. Recall that the objective of this problem is to select a single path for a given source-destination pair such that the overall congestion is minimized. By solving the LP obtained by relaxing the constraints 2.3 one can obtain non-integer values for variables $x(P)$. So it is not clear from the LP solution which path should be selected for a given source-destination pair. A simple randomized rounding scheme for this problem would be to pick exactly one path, P, from the set of paths, P_i, for a given source-destination pair i with probability $x(P)$. In general, *independent randomized rounding* for a 0/1 optimization problem where the variables are binary valued is as follows: if the value of a decision variable x in LP solution is $\alpha \in [0, 1]$, we may set $x = 1$ with probability α, and $x = 0$ with $1 - \alpha$ in the rounded solution. If the objective function is a linear function of the decision variables, it follows immediately that

the expected value of the rounded solution equals the fractional objective value yielded by the LP [44]. We refer interested readers to Raghavan and Thompson [48, 49] and Srinivasan [53] for a more detailed discussion on independent randomized rounding.

There are a couple of challenges in designing an effective rounding technique: (a) for a given instance $\mathcal{I}$ of $\mathcal{P}$, one needs to ensure that after the variables have been rounded the constraints from the linear program are not violated and (b) one needs to show that the rounding results in a feasible solution with objective function value is at least $OPT(\mathcal{I})/\gamma$ for $\gamma \geq 1$ in expectation or with high probability. In this dissertation, for proving the quality of the rounded solution, we exploit an important property of 0/1 independent random variables: when we aggregate a large number of independent 0/1 random variables, their sum is very sharply concentrated around its mean value. More specifically, this property has been captured in the Chernoff bound [13, 39] that states:

Definition 1. *For $X_1, \ldots, X_n$ 0/1 independent random variables such that $Pr(X_i) = p_i$ and $X = \sum_{i=1}^{n} X_i$, and $\mu = E[X]$. The following bounds hold:*

$$for\ 0 < \delta \leq 1, Pr(X \geq (1+\delta)\mu) \leq e^{-\mu\delta^2/3};$$

$$for\ R \geq 6\mu, Pr(X \geq R) \leq 2^{-R}$$

Note that there are different variations of the Chernoff bounds, we refer interested readers to Mitzenmacher and Upfal [39].

In addition to randomized rounding, we use randomization is for breaking ties. In a wireless ad-hoc network setting where there are multiple transmitters (sources) trying to access the same medium, conflicts could arise due to wireless interference if neighboring nodes begin their transmissions at the same time. This could result in packet loss and overall reduction in the system throughput. We use randomization to decide which source should transmit at a given time. We allow each source to wait for a random amount of time before starting its transmission. This technique is based on the *random-delays* technique used by Leighton *et*

al. [31] for wired networks. Our results of Chapter 7 demonstrate that randomization allows us to minimize conflict and bound the end-to-end delay.

2.5 Models and Notation

We now present a description of the models and notation used in this dissertation. Let V denote the set of transceivers located on the Euclidean plane - we will sometimes refer to these transceivers as nodes. Let $d(u, v)$ denote the Euclidean distance between any two nodes $u, v \in V$. For a link $e = (u, v)$, let $\ell(e) = d(u, v)$ denote its length. Let $\mathcal{E} \subseteq V \times V$ denote the set of possible directed links (also referred to as edges), on which transmissions can occur. Let $N_{out}(u)$ denote the set of outgoing links from $u \in V$ and $N_{in}(u)$ denote the set of incoming links to node $u \in V$. Let

$$\Delta = \frac{\max_{e \in \mathcal{E}}\{\ell(e)\}}{\min_{e \in \mathcal{E}}\{\ell(e)\}};$$

where Δ is also called the "length diversity" or maximum inter-point separation [19]. The logarithms used in this work are to the base two unless specified otherwise. The terms such as "$\log \Delta$" should be interpreted as "$1 + \log \Delta$" to ensure that such quantities are bounded away from 0; we do not do this explicitly, in order to avoid notational clutter.

2.5.1 Capacity and Power

The power with which any node $u \in V$ transmits is denoted by $J(u)$. We define

$$\Gamma_{node} = \frac{\max_{u \in V}\{J(u)\}}{\min_{u \in V}\{J(u)\}}.$$

In case different power levels are used on different edges, we denote transmitting power of an edge $e \in \mathcal{E}$ to be $J(e)$. In that case we define

$$\Gamma_{edge} \frac{\max_{e \in \mathcal{E}}\{J(e)\}}{\min_{e \in \mathcal{E}}\{J(e)\}}.$$

The capacity of a link also depends on the system noise and interference caused by other simultaneously transmitting links [1]. In our work, we simplify this assumption by using the Additive White Gaussian Noise (AWGN) model for specifying the link capacities [5]. In this model the capacity $cap(e)$ of a link $e = (u, v)$ is given by

$$cap(e) = W \log_2 \left(1 + \frac{J(e)}{\ell(e)^\alpha N \cdot W}\right), \tag{2.4}$$

where W is the bandwidth, N is the spectral noise density and α is the path-loss exponent In the absence of interference, the above equation provides a theoretical upper bound on the link capacity.

2.5.2 Disk-graph Models

In our work, we consider two different disk-graph models of interference, namely the transmitter and protocol models. We assume that signal propagation is omni-directional. Another standard assumption we make is that the signal decays as $r^{-\alpha}$ with distance r, where $\alpha > 2$ is the path loss exponent [21]. For a graph-based model, we assume that the signal from node u transmitting at power $J(u)$ is received only within a transmission range of $range(u) = (\frac{J(u)}{c_1 N})^{1/\alpha}$, where c_1 is a constant and N denotes the noise spectrum density. It is assumed that a transmission from u to v is possible if $d(u, v) \leq range(u)$.

In the transmitter model (Tx-model) [30] a transmission from node u is successful (i.e. is received correctly by the intended recipient of the transmission) if and only if any other transmitter w is such that, $d(u, w) \geq (1 + \delta) \cdot (range(u) + range(w))$, where $\delta > 0$ is a constant and a parameter of the model (cf. Fig. 2.1).

In the protocol model [58] if a node u transmits to node v, this transmission is successfully received by v if $v \in range(u)$ and for any other transmitting node w, $d(w, v) \geq (1+\delta) \cdot d(u, v)$, where δ is a constant and a parameter of the model.

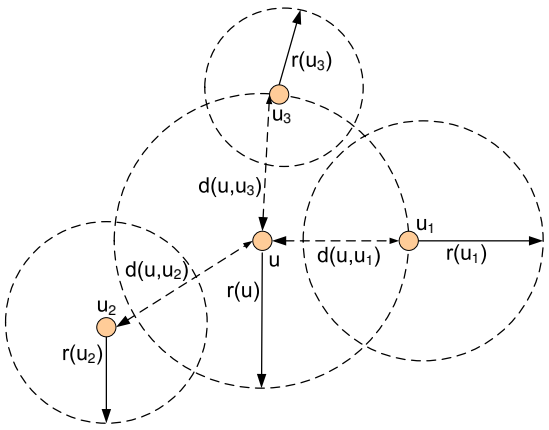


Figure 2.1: Example of the Tx interference model: The transmission range associated with each node is represented by a dotted circle, centered at the node. Dotted lines represent the distance between two nodes. Let δ be a small positive constant. It can be seen that $d(u, u_1) \leq (1 + \delta) \cdot (range(u) + range(u_1))$. This is also true for nodes u_2, u_3, therefore transmissions on nodes u_1, u_2, u_3 interfere with the transmission on node u.

2.5.3 The SINR Interference Model

We use the *Signal-to-Interference-plus-Noise-Ratio* (SINR) model of interference as described in [40]. In this setting, for any given link $e = (u, v)$ with u as the sender and v as the receiver, the received signal power, $J_v(e)$, at node v due to sender u, can be expressed as $J_v(e) = \dfrac{J(e)}{d(u,v)^\alpha}$, where α denotes that path-loss exponent. As suggested in [42] we assume that $\alpha > 2$.

For a given set $E' = \{e_i = (u_i, v_i), i \in \{1, \ldots, k\}$ of simultaneously communicating links, the interference $I_r(v_i)$ at receiver v_i from all the senders can be expressed as

$$I_r(v_i) = \sum_{e_j=(u_j,v_j)\in E', e_j \neq e_i} \frac{J(e_j)}{d(u_j, v_i)^\alpha}.$$

The node v_i can successfully receive the transmission from u_i, if

$$SINR(v_i) = \frac{J(e_i)}{d(u_i, v_i)^\alpha [N + I_r(v_i)]} \geq \beta,$$

where β denotes the antenna gain and N is the spectral noise density. It should be noted that in order to calculate the SINR at a given receiver v_i, we consider the transmission power on the link $e_i = (u_i, v_i)$, because a node u_i could potentially transmit at different power levels on different outgoing links from the set $N_{out}(u_i)$. For the case where all the nodes u_i transmit at a fixed power level $J(u_i)$, the SINR at any given receiver v_i can be expressed as:

$$SINR(v_i) = \frac{J(u_i)}{d(u_i, v_i)^\alpha [N + \sum_{e_j=(u_j,v_j)\in E', u_j \neq u_i} \frac{J(u_j)}{d(u_j, v_i)^\alpha}]} \geq \beta.$$

2.5.4 Congestion and Dilation

We define a quantity called *congestion* C as follows.

Definition 2.

$$\begin{aligned} \forall e = (u, v) \in \mathcal{E},\ C(e) &= \{e' = (u', v') \in \mathcal{E} : \\ &\quad a \cdot d(u', v') \geq d(u, u') \bigwedge d(u', v') \geq d(u, v)\} \\ C &= \max_{e \in \mathcal{E}} |C(e)|. \end{aligned}$$

Here, a is a constant such that $a \geq 4\sqrt[\alpha]{\frac{48\beta(1+\epsilon)}{\epsilon(\alpha-2)}}$, where ϵ is a small positive constant having a value greater than 0, and $\alpha > 2$ [42]. The congestion for a given edge e captures the number of transmissions in the vicinity of e. Figure 2.2 demonstrates the congestion measure for link $e = (u, v)$. Our definition of congestion is motivated by the one discussed in [29, 31].

For a given set of source-destination pairs, $K = \{(s_1, t_1), \ldots, (s_k, t_k)\}$, let P_i be the path selected for connection i and let $\mathcal{P} = \{P_1, \ldots P_k\}$ be the set of all paths. We define dilation $D(\mathcal{P})$ as the length of the longest path in $\mathcal{P}$.

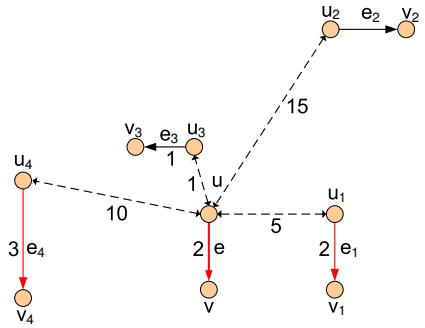


Figure 2.2: Illustration of congestion measure for a link. The solid lines represent edges along with edge lengths (e.g. $l(e) = 2$). The dotted lines represent distance between two nodes (e.g. $d(u, u_1) = 5$). Let $a = 6$. Then $C(e) = \{e_1, e_3\}$ for link $e = (u, v)$. By definition, $e_4, e_2 \notin C(e)$ since $a \cdot d(u_4, v_4) < d(u, u_4)$ and $d(u_2, v_2) < d(u, v)$.

V, $n = \|V\|$	set and number of nodes
$\mathcal{E} = \{(u, v) \in V \times V\}$	set of all edges
$\mathcal{K} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$	k source-destination pairs
$\mathcal{T}$	Set of all possible edge-power level transmissions
$\mathcal{Q} = \{Q_1, \ldots, Q_k\}$	set of packets for k sources
$\mathcal{P} = \{P_1, \ldots, P_k\}$	set of paths for k connections
$E = E(\mathcal{P})$	set of edges on paths $\mathcal{P}$
$C(e)$	congestion set for link e
C congestion	max. cardinality of any $C(e)$
$D =$ dilation	length of longest path in $\mathcal{P}$
$[j_{min}, j_{max}]$	minimum and maximum power levels
$\bar{J}_{edge} = \{J(e_1), \ldots, J(e_n)\}$	set of power assignments for links
$\mathcal{J}$	set of possible power levels to choose from
Γ_{node}	ratio of max. to min. power assignments for every node
Γ_{edge}	ratio of max. to min. power assignments for every link
Δ	ratio of max. to min. node-distances
$L = \{0, \ldots, \lfloor \log \Delta \rfloor\}$	index set of possible edge lengths
$M = \{0, \ldots, \lfloor \log \Gamma_e \rfloor\}$	index set of possible power levels
$range(u)$	Range for u at power $J(u)$
$cap(e)$	capacity of link e
$\mathcal{S}$	a valid schedule
$N_{out}(u)$	set of outgoing edges from u
$N_{in}(u)$	set of incoming edges to u
$I(e)$	Interference set for link e
$I(e, p)$	Interference set for transmission e at power p
$I_{\geq}(e)$	set containing $e' \in I(e), \ell(e') \geq \ell(e)$
$I_{\geq}(e, p)$	set containing $(e', p') \in I(e, p), p' \geq p$
$f_i(e)$	flow on edge e for connection i
$f_i(e, p)$	flow on transmission (e, p) for connection i
$f(e) = \sum_{i=1}^{k} f_i(e)$	total flow on edge e
$f(e, p)$	total flow on transmission (e, p)
r_i	rate for each connection i
$r = \sum_i^k r_i$	total rate
$x(e) = \frac{f(e)}{cap(e)}$	link utilization
$x(e, p)$	link utilization for transmission (e, p)
W	single window or frame
α	path-loss exponent
β	antenna gain
N	ambient noise
B	bound on total power consumed
$a, \lambda_0, \lambda_1, \lambda_2, \epsilon, c_1$	positive constants

Table 2.1: List of notation

Chapter 3

Literature Review

In this chapter, we present an extensive review of the literature in the area of capacity characterization of wireless networks.

An influential work on capacity of wireless networks has been by Gupta and Kumar [21]. They show that, given n identical randomly distributed nodes in the unit square, with each node having an independent randomly chosen destination, the uniform per node throughput capacity in bit-meters/second, is $\theta(\frac{1}{\sqrt{n \log n}})$ for the protocol interference model. This result is a scaling law and indicates that as the number of nodes in the system increases, the per-node capacity does not scale adequately. The asymptotic performance bounds presented in [21] hold for the case of random networks, and cannot be extended for any given network with a specified workload. Following the work of Gupta and Kumar [21], many researchers have studied various aspects of capacity of wireless networks.

Li *et al.* [33] study the impact of traffic patterns on the achievable capacity and show that as the network grows in size, if the average distance between the source and the destination remains relatively small, the per-node capacity stays roughly constant. They demonstrate that a random traffic pattern represents the worst case from the viewpoint of per-node capacity. The authors also consider the interactions of the 802.11 medium access control

protocol and ad-hoc forwarding. They show that the 802.11 scheme performs satisfactorily in scheduling packets in ad-hoc networks, and is able to approach the per-node capacity bound derived by Gupta and Kumar in [21]. The work by Grossglauser and Tse [20] examines the asymptotic throughput capacity of large mobile wireless networks and shows that, for delay tolerant systems, per-node throughput capacity can increase significantly when nodes are mobile rather than fixed. The authors utilize mobility to keep data transfers local (i.e. packet transmission occurs between any two nodes only when they are close to each other, thereby reducing the total interference) and achieve a $O(n)$ throughput capacity. The authors however impose weak constraints on the overall delay and acknowledge that their methods could potentially lead to significant packet transmission delays.

Yi *et al.* [58] investigate the throughput capacity of ad-hoc networks using directional antennas. Directional antennas as opposed to omnidirectional antennas can allow more than one pair of nodes located in each other's vicinity to communicate simultaneously. This can increase the spatial reuse of the wireless channel and reduce radio interference. The authors show that for a random network, the throughput capacity with the use of directional antennas can achieve a gain as large as $\frac{4\pi^2}{\alpha\beta}$, where α, β are angles of the transmitter and receiver respectively.

The work by Liu *et al.* [38] studies the throughput capacity of hybrid networks formed by placing base stations in an ad-hoc network. The base stations are assumed to have a high-bandwidth wired backbone. The authors show that if the number m of base stations grows faster than $\sqrt{n}$, the throughput capacity increases with the number of base stations.

In addition to extending the work of [21] and finding asymptotic performance bounds, there has been a significant amount of research in utilizing various approximation techniques and cross-layer protocols for maximizing the throughput of a given multi-hop wireless network with a specified workload. We shall classify the literature belonging to this area based on the approximation techniques used.

3.1 Linear Programming techniques

The throughput capacity maximization problem has been formally studied and proved to be NP-hard by Jain *et al.* [22] for the disk-graph interference model. The authors use a linear programming based approximation to characterize the throughput capacity of the network. They model interference constraints as a *conflict graph* and provide techniques for estimating the upper and lower bounds on the overall network capacity. Buragohain *et al.* [7] point out that the techniques discussed in Jain *et al.* [22] could have an exponential complexity.

Kodialam and Nandagopal [25] consider the problem of determining the achievable rates for multi-hop wireless networks by simultaneously performing routing and scheduling. The authors provide tight necessary and sufficient conditions for link flow stability and develop a $\frac{2}{3}$ factor approximation algorithm and a greedy coloring algorithm for solving the routing and scheduling problem respectively. The interference model used in [25] is extremely simplistic, in which only the primary interference constraints (where a node can transmit or receive from at most one node at a time) are imposed. In [26], the authors extend their results for the case of full and half duplex orthogonal channels and develop a $\frac{1}{2}$ approximation algorithm for the routing problem. The interference model used in [26] handles the case where a node can have multiple receive channels and can receive transmissions from multiple neighbors simultaneously.

Recent work by Kumar *et al.* [30] provides a constant-factor approximation algorithm for the throughput maximization problem with joint scheduling and routing constraints for a single channel, single radio network. The results provided in [30] are based on a linear programming (LP) formulation that captures the properties of wireless interference models. The results hold for various disk-graph interference models such as protocol, transmitter, transmitter-receiver model and the approximation ratio depends on the type of interference model used.

Buragohain *et al.* [7] extend the work of [30] and achieve lower bounds on the total throughput

achieved for a grid network. They develop a node-based linear programming formulation that provides an improved constant factor approximation to the total throughput. The interference model considered by [7] is a variation of the disk-graph model, in which any two links interfere with each other if they lie in each others transmission disks.

3.2 Primal-Dual, Convex Optimization Algorithms and Backpressure Schemes

Lin and Shroff [35] study the problem of joint rate allocation and scheduling. They use a dual optimization based approach to decompose the problem into rate control and scheduling problem. Their technique provides an optimal solution that maximizes the throughput and provides a stable and fair schedule. In [36], the authors improve the performance of their scheduling scheme by considering sub-optimal solutions.

Cruz and Santhanam [17] study the problem of joint routing, link scheduling and power control to achieve high data rates for multi-hop wireless networks. The authors first present a duality based algorithm for finding an optimal link scheduling and power control policy to support a given traffic rate on each link in the network. Next, they construct a policy to allocate traffic rates on each link using a shortest path based routing algorithm. The algorithm presented in [17] suffers from a worst-case exponential complexity as the number of transmission nodes increase.

Recent work by Chen *et al.* [12] considers the problem of congestion control and resource allocation (via routing and scheduling). The paper aims to maximize resource allocation by using a duality-based approach to decompose the main problem into sub-problems of congestion control, routing and scheduling. The authors further extend the dual algorithms to take into account time-varying channels and adaptive multi-rate devices.

Bhatia and Kodialam [5] study the joint routing, scheduling and power control problem with

the objective of minimizing the total power consumed. They formulate the overall problem as a non-convex optimization problem with non-linear constraints and develop a polynomial time 3-approximation algorithm for solving this problem. The authors consider the impact of primary interference constraints.

Kodialam and Nandagopal [27] consider the problem of characterizing the capacity regions of multi-radio, multi-channel wireless mesh networks (MCMR). The authors provide necessary conditions to verify the feasibility of rate vectors in MCMR networks and derive upper bounds on the capacity using a fast primal-dual algorithm. They further develop link channel-assignment schemes to derive lower bounds on the achievable throughput capacity.

Tassiulas and Ephremides [55] study the problem of scheduling in multi-hop wireless networks for known packet arrival rates at every source. They specify a scheduling policy and prove that this policy results in stable schedules (bounded queue sizes) for known arrival rates. Building on this work, recent work by Neely [43] considers the problem of performing joint routing and scheduling using a backpressure routing algorithm for maximizing network throughput and minimizing average power utilized. The author considers a multi-node, multi-hop wireless network with unreliable channels having high error probabilities. In the backpressure scheme proposed, each transceiver maintains a backpressure index that compares its current queue backlog with that of its one-hop neighbors. A potential next-hop receiver is then selected based on this backlog index and probability of successful transmission from the transmitter to the receiver. The stability of the scheduling scheme is derived using the Lyapunov theory [55].

In a similar line of research, Chaporkar *et al.* [11] attempt to characterize the maximum throughput region for arbitrary networks via a distributed *maximal* scheduling scheme. By imposing constraints on the stochastic arrival rates, the authors provide performance guarantees and ensure the stability of the scheduling scheme. In [10], the authors demonstrate that the maximal scheduling technique proposed in [11] ensures queue-length stability in which the queue-lengths remain bounded at all times. The techniques developed in [10, 11, 43] can

incorporate variable arrival rates. However these techniques mainly work for the case of uniform power-levels and graph-based interference models.

3.3 Scheduling with the SINR model

Recent work by Moscibroda *et al.* [40] studies the problem of scheduling packets with SINR constraints to ensure strong-connectedness of the graph. The authors study the scheduling complexity of connecting a given number of nodes located at arbitrary positions by a communication tree and present a scheduling algorithm that successfully schedules a strongly connected set of links in poly-logarithmic time. Subsequent work by Moscibroda *et al.* [42], studies the MAC level scheduling problem with SINR constraints. The authors present a scheduling algorithm that successfully schedules a set of links in poly-logarithmic time by assigning non-uniform power to the transmitting nodes.

In a recent study by Brar *et al.* [6], the authors present a polynomial time heuristic for computing a feasible schedule with the SINR interference model and derive an approximation factor for the length of the schedule for random distribution of nodes.

3.4 Randomized Algorithms

Kumar *et al.* [29] develop a randomized poly-logarithmic factor approximation algorithm for minimizing end-to-end latencies. They develop distributed protocols for routing and scheduling for various families of disk-graphs.

Recent work by Lin and Rasool [34] examines the performance of synchronous random-access for two types of interference node-exclusive and the two-hop interference model. In the node-exclusive model, links interfere with each other only if they share a common end-point. The authors design a random-access scheduling strategy based on the notion of

periodic contention-frames, which is guaranteed to achieve a $\frac{1}{3}$-factor of the optimal capacity region of the network. Joo and Shroff [23] propose an improved strategy for the node-exclusive interference model that achieves a 1-factor of the optimal capacity region. Both of these works also study the two-hop interference model and propose synchronous random-access schemes that achieve a $\Delta + 1$-factor of the optimal capacity, where Δ is the network interference degree (maximum number of links that interfere with some specific link, but are mutually interference-free amongst themselves).

Recent work by Chafekar *et al.* [9] studies the capacity of wireless networks that utilize asynchronous random-access scheduling. A distributed randomized protocol is developed that allows each link to access the wireless channel with a certain probability. These link transmission probabilities are chosen such that the achieved throughput capacity is provably close to the optimal solution.

Chapter 4

Throughput Maximization with SINR constraints

4.1 Problem Description

We consider the problem of maximizing overall throughput with the SINR interference model (TM-SINR). The formal description of the problem is as follows: Given (i) a set of nodes V and a set $\mathcal{E} = V \times V$ of possible directed edges, (ii) a set of source-destination pairs $\mathcal{K} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$, and (iii) a power level vector $\bar{J}_{edge}$ that specifies power level $J(e)$ for transmission on edge $e \in \mathcal{E}$, the objective of TM-SINR problem is to maximize total throughput capacity by: (i) choosing routes for the connections, (ii) choosing data rates on the routes, and (iii) scheduling the packets at each time slot, such that SINR constraints are satisfied for all simultaneous transmissions. Note that the TM-SINR problem does not involve power control, i.e., the power level $J(e)$ for each edge is fixed and is given as part of the input.

The SINR constraints make the throughput optimization problem *non-convex*. Further the link scheduling problem with SINR constraints has been shown to be NP-hard in [19]. Since

scheduling is also an integral component of our problem, it is reasonable to conjecture that the TM-SINR problem is also NP-hard. We therefore focus on developing rigorous polynomial time algorithms with provable performance guarantees.

4.2 Overview of Results

The main results obtained for the TM-SINR problem are summarized as follows:

- We construct network instances and compare the throughput predicted by the SINR and disk-graph model for these instances. We observe that the predicted throughput capacity is significantly different in these two models. For the case of uniform power levels (the case when power level for all edges is the same), we show that there are instances in which throughput capacity predicted by the SINR model is significantly higher than that predicted by the disk-graph model. For the case of linear power levels (the case when power level for each edge is $J(e) \propto \ell(e)^{\alpha}$, where $\ell(e)$ denotes the edge length), we show that there are instances in which throughput capacity estimated by the disk-graph model is higher than that estimated by the SINR model.

- We develop a linear programming based approach to approximate the maximum throughput rate vector in the case of SINR constraints. For the general case of non-uniform power levels, in which the power levels on different edges could be different, we develop a *polynomial time* approximation algorithm that provides a feasible rate vector whose total throughput is at least $O(r_{opt}/\log\Delta \cdot \log\Gamma_{edge})$, where r_{opt} denotes the maximum possible throughput, Δ denotes the maximum inter-point separation and Γ_{edge} is the ratio between maximum and minimum power levels respectively (cf. Chapter 2, Table 2.1). Recall form Chapter 2 that "$\log\Delta$" should be interpreted as "$1 + \log\Delta$" in order to ensure that such quantities are bounded away from 0. Our approximation bound is a worst case guarantee that holds for every instance.

- We consider two special cases of power level choices - *uniform* power level choice and *linear* power level choice - we guarantee that the throughput is at least $O(r_{opt}/\log\Delta)$.

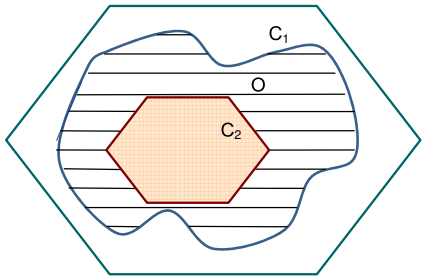


Figure 4.1: Linear Relaxation: The outer most region C_1 corresponds to the rate region satisfying the necessary conditions. The middle region O represents the feasible rate region. The innermost region C_2 corresponds to the rate region satisfying the sufficient conditions

4.3 Overall Approach

We develop a general technique for solving the TM-SINR problem. Following are the main components of our technique:

- *Linear Relaxation*: We first mathematically formulate the TM-SINR problem. The exact formulation yields a *non-convex* rate region due to the non-linearities arising from the SINR model and the scheduling constraints. We develop a linear program by *relaxing* the scheduling constraints. We extend the *relaxation* techniques discussed in [30] for the TM-SINR problem.

- *Necessary Conditions*: We exploit the geometric nature of the problem to derive the necessary conditions. These conditions given an upper bound on the rate region (cf. Figure 4.1).

- *Sufficient Conditions and Scheduling Algorithm*: The rate vectors corresponding to the necessary conditions could violate the actual interference constraints and could result in an infeasible schedule. We therefore derive the sufficient conditions by uniformly scaling down the necessary conditions. These conditions given lower bound on the rate region (cf. Figure 4.1). We develop a greedy scheduling scheme and prove that the rate vectors corresponding to the sufficiency conditions can be scheduled feasibly (without any violations of the SINR constraints) using this scheme.

4.4 Notation

In this section, we briefly describe the notation used in this chapter. We consider the input instance of the TM-SINR problem to be specified as $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$, where V denotes a set of transceivers, $\mathcal{E} \subseteq V \times V$ denotes the set of possible directed links on which transmissions can occur, $\mathcal{K}$ is a set of connections, with the ith connection from node s_i to node d_i, and $\bar{J_{edge}} = (J(e) : e = (u, v) \in \mathcal{E})$ specifies the vector of power transmission levels on edges. We recall the definitions of $N_{out}(u), N_{in}(u)$, Δ, Γ_{edge}, congestion C, SINR model, and constants α, β, a from Chapter 2 and Table 2.1. Let $L = \{0, \ldots, \lfloor \log \Delta \rfloor\}$ and $M = \{0, \ldots, \lfloor \log \Gamma_{edge} \rfloor\}$. Let $j_{main} = \min_{e \in \mathcal{E}} J(e)$. Without loss of generality, we assume that $\min_{e \in \mathcal{E}}\{\ell(e)\} = 1$. We define $H_k^i = \{e \in \mathcal{E} : \ell(e) \in [2^i, 2^{i+1}) \wedge J(e) \in [j_{min} \cdot 2^k, j_{min} \cdot 2^{k+1})\}$, for $i \in L, k \in M$. The (long term) fairness index $\lambda \in [0, 1]$ is defined to be the ratio between the minimum and maximum rates, i.e. $\lambda = \dfrac{\min_i r_i}{\max_i r_i}$.

Link rates and feasible schedules: For the schedule constructed in this work, we assume that the overall time is divided into slots of uniform length τ. We also assume that the system operates in a synchronous mode (i.e. transmissions occur at the beginning of a time slot) and the transmissions occur on an error-free channel. Let $f_i(e)$ denote the mean flow rate on link e for the ith connection, and let $f(e) = \sum_i^k f_i(e)$ denote the total link flow. We let $x(e) = f(e)/cap(e)$ denote the *link utilization* - this denotes the fraction of time link e is

used. The vectors $\bar{f}$ and $\bar{x}$ are called the link flow and *link utilization* vectors respectively. An end-to-end schedule $\mathcal{S}$ describes the specific times at which packets are transmitted over the links of the network. For schedule $\mathcal{S}$, let $X(e,t)$ be an indicator variable such that

$$X_{e,t} = \begin{cases} 1 & \text{if } e \text{ transmits with at time } t \\ 0 & \text{otherwise.} \end{cases}$$

We say that $\mathcal{S}$ is valid if the SINR constraints are satisfied at all the receivers at any time t. We say that $\mathcal{S}$ feasibly schedules the *link utilization* vector $\bar{x}$ if we have $\lim_{T\to\infty} \sum_{t\leq T} \frac{X(e,t)}{T} = x(e)$ for each edge e - in this case, we say that $\mathcal{S}$ corresponds to the utilization vector $\bar{x}$. The rate region $\mathcal{X}(\mathcal{I})$ is the space of all utilization vectors $\bar{x}$ for the instance $\mathcal{I}$ of TM-SINR that can be scheduled feasibly. Let r_i denote the end-to-end rate on the ith connection in bits per second, resulting from the flow vector $\bar{f}$. In this chapter we are interested in maximizing the total end-to-end rate $\sum_i^k r_i$. For an instance $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$ of TM-SINR, let $r_{opt}(\mathcal{I})$ denote the maximum possible total throughput rate that is feasible.

4.5 SINR vs Disk-graph models

In this section, we compare the SINR and disk-graph (also known as graph-based) models in the context of the throughput maximization problem. Given an instance $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J_{edge}})$ of TM-SINR, we follow the approach of [42] in constructing an "equivalent" connectivity graph $G = (V, \mathcal{E}_{gm})$ and a resulting instance $\mathcal{I}_{gm}$ in a disk-graph model in the following manner. In the rest of this section, we will consider instances $\mathcal{I}$ of TM-SINR in which every node $u \in V$ uses a fixed power level $J(u) = J$ for every incident link $e = (u, v) \in \mathcal{E}$. We associate a transmission range of $r(u) = (J(u)/c_1)^{1/\alpha}$ with every node $u \in V$ for a constant c_1, giving rise to a disk graph $G = (V, \mathcal{E}_{gm})$ with $(u, v) \in \mathcal{E}_{gm}$ if $d(u, v) \leq r(u)$. This is a directed graph in general, if nodes have non-uniform transmission ranges. The corresponding instance $\mathcal{I}_{gm}$ consists of this graph G along with the same set $\mathcal{K}$ of connections, as in $\mathcal{I}$. Note that the set of edges on which transmissions can happen is the same in both models.

For every edge $e \in \mathcal{E}_{gm}$, we use the same expression for $cap(e)$, the capacity of edge e as in $\mathcal{I}$, since this comes from the AWGN model. What is different is the interference - we can now use any graph-based interference model to specify the set $I_{gm}(e)$ of edges that interfere with e - for concreteness, we use the distance-2 matching model [30], which defines $I_{gm}(e) = \{e' = (u', v') : d_G(\{u, v\}, \{u', v'\}) \leq 1\}$, where $d_G()$ defines the distance between two sets in the graph G. A schedule is valid in the disk-graph model, if at any time, no edge e is simultaneously scheduled along with some edge $e' \in I_{gm}(e)$. Let $r^{gm}_{opt}(\mathcal{I}_{gm})$ denote optimum throughput rate possible for instance $\mathcal{I}_{gm}$ in the disk-graph model.

We show the following results in this Section.

- If the instance $\mathcal{I}$ of TM-SINR has *uniform* power levels, the ratio $r_{opt}(\mathcal{I})/r^{gm}_{opt}(\mathcal{I}_{gm})$ can be arbitrarily large, i.e., the corresponding graph-based model underestimates the throughput capacity significantly.
- In contrast, when the power levels in the instance $\mathcal{I}$ of TM-SINR are linear, we show that the ratio $r_{opt}(\mathcal{I})/r^{gm}_{opt}(\mathcal{I}_{gm})$ can be arbitrarily small.

The above results show that if the power levels are fixed, the total throughput in both the models is very different.

4.5.1 Uniform power levels

We construct the following instance $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$ of TM-SINR, with *uniform* power level J for all transmissions. Let $R = (J/c_1)^{1/\alpha}$ be the corresponding transmission range in the corresponding graph model, as discussed earlier; we assume that R is a large integer. Let $V = \{u_0\} \cup_{i=1}^{n} \{u_i, v_i\}$ be a set of nodes, which are placed in the following manner. Imagine a circle of radius $R/2$ centered at node u_0, and the nodes $u_1, \ldots, u_n$ uniformly placed on the circumference of this circle at a spacing of $\Theta(\sqrt{R})$, so that $n = \Theta(\sqrt{R}) = c_2 \cdot \sqrt{R}$ (cf. Figure 4.2). Each v_i is at a unit distance from u_i, for $i = 1, \ldots, n$. Let the connections in $\mathcal{K}$ in the

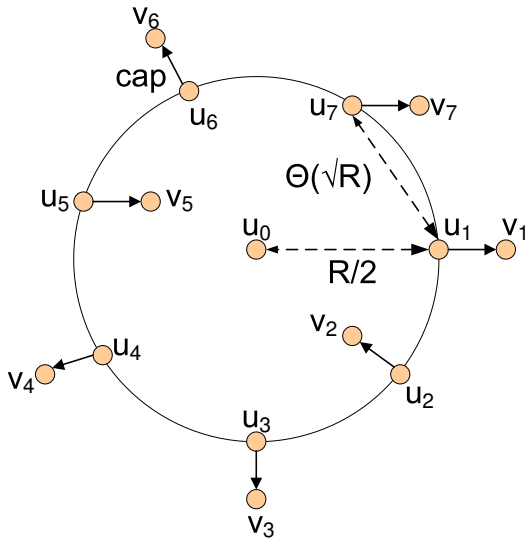


Figure 4.2: Illustrating an example that compares the throughput achieved between SINR and graph- based models for *uniform* power levels. A circle of radius $R/2$ is centered at node u_0. Distance between any two adjacent u_i, u_j is $\Theta(\sqrt{R})$. Each edge $e_i = (u_i, v_i)$ has length $\ell(e_i) = 1$ units and $J(e_i) = J$.

instance $\mathcal{I}$ be all the pairs $e_i = (u_i, v_i)$, for all $i = 1, \ldots, n$. Let $cap = cap(e_i)$ denote the capacity of any link e_i in bits/sec; note that this is the same for every edge e_i in this setting. In order to simplify analysis we ignore the ambient noise, i.e., assume $N_0 = 0$. The results can be extended to take the noise into account.

Lemma 4.1. *For the instance $\mathcal{I}$ of* TM-SINR *and the corresponding graph-based instance $\mathcal{I}_{gm}$ described above, we have $r_{opt}(\mathcal{I})/r^{gm}_{opt}(\mathcal{I}_{gm}) = \Omega(cap \cdot \sqrt{R})$, assuming $\beta \leq c_4 \cdot R^{(\alpha-1)/2}$, for a constant c_4.*

Proof: Observe that for all $i \neq j$, $\sqrt{R} \leq d(u_i, u_j) \leq R$. Therefore, $I_{gm}(e_i) = \{e_j : j \neq i\}$, which implies that at any time, at most one edge e_i can be scheduled in the graph-based model in the instance $\mathcal{I}_{gm}$. This implies that $r^{gm}_{opt}(\mathcal{I}_{gm}) = \Theta(cap)$ bits/sec.

Next, consider the SINR model for the instance $\mathcal{I}$ of TM-SINR. Suppose all the edges $e_i = (u_i, v_i)$ are scheduled simultaneously - the SINR ratio at any receiver v_i in this case is,

$$\begin{aligned} SINR(v_i) &= \frac{J}{\ell(e_i)^\alpha \left[\sum_{j \neq i}^{n} J/d(v_i, u_j)^\alpha\right]} \\ &\geq \frac{c_1 R^\alpha}{\left[c_2 \cdot \sqrt{R} c_1 \cdot R^\alpha/(c_3 \cdot \sqrt{R})^\alpha\right]} \\ &\geq \beta \end{aligned}$$

where the first inequality follows from the fact that $J = c_1 \cdot R^\alpha$, $n = c_2 \cdot \sqrt{R}$, and $d(v_i, u_j) \geq (\sqrt{R}) = c_3 \cdot (\sqrt{R})$ for this instance, and the second inequality follows if $\beta \leq c_4 \cdot R^{(\alpha-1)/2}$ for a constant $c_4 = c_3^\alpha/c_2$. This implies that all the edges e_i can be scheduled simultaneously in the SINR model, leading to $r_{opt}(\mathcal{I}) = \Theta(cap \cdot \sqrt{R})$, and so the Lemma follows.

4.5.2 Linear Power Levels

We now construct an instance $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$ of TM-SINR with *linear* power levels, i.e., for each $e \in \mathcal{E}$, $J(e) = c_1 \ell(e)^\alpha$, for constant c_1. The set $V = \cup_{i=1}^{n} \{u_i, v_i\}$ has $2n$ nodes, which are located on a line in the order $u_1, v_1, u_2, v_2, \ldots, u_n, v_n$. For all $i = 1, \ldots, n$,

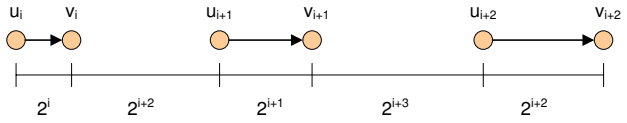


Figure 4.3: Illustration of an example that compares the throughput achieved between SINR and graph- based models for *linear* power levels. A line topology is constructed, such that each edge $e_i = (u_i, v_i)$ has length $\ell(e_i) = 2^i$ and $d(v_i, u_{i+1}) = 2^{i+2}$. $J(e_i) = c_1 \ell(e_i)^\alpha$, for constant c_1.

we have $d(u_i, v_i) = R_i = 2^i$, and for all $i = 1, \ldots, n-1$, we have $d(v_i, u_{i+1}) = 2^{i+2}$ (cf. Figure 4.3). The set $\mathcal{E} = \{e_i = (u_i, v_i) : i = 1, \ldots, n\}$ will be the only edges used for transmission, with $J(e_i) = c_1 R_i^\alpha$, for each i. All the connections in $\mathcal{K}$ in this instance are the pairs $e_i = (u_i, v_i)$, for $i = 1, \ldots, n$. Because of our AWGN model for the link capacities (cf. Chapter 2, Equation 2.4), it follows that for all $e_i \in \mathcal{E}$, $cap(e_i) = cap$ is a fixed value. Each node u_i has only one incident edge in the set $\mathcal{E}$, so for the graph-based model, we set $r(u_i) = R_i = 2^i$. Therefore, for the corresponding graph-based instance $\mathcal{I}_{gm}$, the connectivity graph $G = (V, \mathcal{E}_{gm})$ has $\mathcal{E}_{gm} = \{e_i : i = 1, \ldots, n\}$.

Lemma 4.2. *For the instance $\mathcal{I}$ and the corresponding graph-based instance $\mathcal{I}_{gm}$ described above, we have $r^{gm}_{opt}(\mathcal{I}_{gm})/r_{opt}(\mathcal{I}) = \Theta(n)$.*

Proof: First, observe that for the graph-based interference in the instance $\mathcal{I}_{gm}$, we have $I_{gm}(e_i) = \phi$ for each $e_i \in \mathcal{E}_{gm}$. Therefore, the edges e_i do not interfere with each other and all these edges can transmit simultaneously in this model, leading to a throughput capacity of $\Omega(n \cdot cap)$.

Next, consider the SINR model. For simplicity, we ignore the noise density N_0. Let E' be any subset of these edges that can transmit simultaneously, and let e_i be the shortest edge among them. For all $e_j \in E'$, $e_j \neq e_i$, we have $d(u_j, v_i) \leq \sum_{k=i}^{j-1}(2^{k+2} + 2^{k+1}) \leq c_2 2^j = c_2 R_j$,

for constant c_2. For these transmissions to be feasible in the SINR model, we must have

$$\frac{J(e_i)}{\ell(e_i)^\alpha \left[\sum\nolimits^{|}_{e_j \in E', e_j \neq e_i} E'| \frac{J(e_j)}{d(u_j,v_i)^\alpha}\right]} \geq \beta,$$

where the LHS is the SINR ratio at v_i. Rearranging, and using the fact that $d(u_j, v_i) \leq c_2 R_j$ for each $e_j \in E'$, we have $|E'|$ is $O(1/\beta)$, which is a constant.
This implies $r_{opt}(\mathcal{I}) = O(cap/\beta)$, and so the Lemma follows.

4.6 TM-SINR for Generic Power Levels

In this Section, we consider the general case of power levels, wherein the power level on every edge $e \in \mathcal{E}$ is $J(e)$ and is specified by the corresponding vector $\bar{J}_{edge}$.

4.6.1 Problem Formulation

In this Section we mathematically formulate the TM-SINR problem. We consider input instances of TM-SINR specified as $\mathcal{I} = (V, E, \mathcal{D}, \bar{J}_{edge})$. Due to the non-linearity of the SINR constraints, the exact formulation of the TM-SINR problem is *non-convex*. We develop a linear programming *relaxation* of this problem by combining the approaches of [8, 30] - we show that both **necessary** and **sufficient** conditions can be derived for the feasible rate region by considering the total *link utilization* in the edges in the set $C(e)$ for any edge e. In order to achieve a *stable* and *feasible* schedule, we partition the set E of edges into sets $H^i_k = \{e = (u, v) \in E : \ell(e) \in [2^i, 2^{i+1}) \wedge \ J(e) \in [j_{min} \cdot 2^k, j_{min} \cdot 2^{k+1})\}, \forall i \in L, k \in M\}$. As we discuss in the next sub-sections, this partitioning helps the scheduling algorithm to bound the number of links that can be scheduled simultaneously without violating the SINR constraints at every receiver. Our formulation for instance $\mathcal{I}$ described below is denoted by $\mathcal{P}(\eta, \mathcal{I})$, where η is a parameter.

$$\max \sum_{i \in \mathcal{K}} r_i \text{subject to:}$$

$$\forall i \in \mathcal{K}, r_i = \sum_{e \in N_{out}(s_i)} f_i(e) \tag{4.1}$$

$$\forall i \in \mathcal{K}, \sum_{e \in N_{in}(s_i)} f_i(e) = 0 \tag{4.2}$$

$$\forall e \in \mathcal{E},\ x(e) = \sum_{i \in \mathcal{D}} f_i(e)/cap(e) \tag{4.3}$$

$$\forall i \in \mathcal{K}, \forall u \neq s_i, t_i, \sum_{e \in N_{out}(u)} f_i(e) = \sum_{e \in N_{in}(u)} f_i(e) \tag{4.4}$$

$$\forall e \in \mathcal{E}, \forall i \in L, \forall k \in M, \sum_{e' \in C(e) \cap H_k^i} x(e') \leq \eta \tag{4.5}$$

$$\forall i \in \mathcal{C}, \forall j \in \mathcal{C} \backslash \{i\}, r_i \geq \lambda \cdot r_j \tag{4.6}$$

In the above formulation, constraints (4.1) define the total rate r_i for each connection, constraints (4.2) define the link utilization $x(e)$ for each link e, constraints (4.3, 4.4) ensure flow conservation, and constraints (4.5) are relaxed congestion constraints - these are the *key* constraints that allow us to use this program to derive upper and lower bounds on the optimum rate. Equation 4.6 captures the fairness constraints that ensure that all the flows obtain a fair share of the throughput. $\lambda = 0$ implies that there is no fairness in the system and $\lambda = 1$ means that all that all the flows have identical throughput. The program $\mathcal{P}(\eta, \mathcal{I})$ has *polynomial* size and can be solved in *polynomial time*.

In the subsequent Sections, we show that the optimum utilization vector satisfies $\mathcal{P}(\eta, \mathcal{I})$ for some constant value of η. We then show that scaling the constraints down by a factor of η allows us to schedule the flows feasibly.

4.6.2 Link-Flow Scheduling: Necessary Conditions

In this Section we derive an upper bound on $r_{opt}(\mathcal{I})$ for a suitable choice of η.

Lemma 4.3. *Let $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$ be an instance of the* TM-SINR *problem, and let $\bar{x} \in$*

$\mathcal{X}(\mathcal{I})$ *be any* feasible link utilization *vector. Then,* $\bar{x}$ *satisfies the following conditions:*

$$\forall e \in \mathcal{E}, \forall i \in L, \forall k \in M, \sum_{e' \in C(e) \cap H_k^i} x(e') \leq \eta_0,$$

where $\eta_0 = \theta \frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2$, *and* $a \geq 4\sqrt[\alpha]{\frac{48\theta\beta(1+\epsilon)}{\epsilon(\alpha-2)}}$. *This implies that* $\bar{x}$ *is a feasible solution to the program* $\mathcal{P}(\eta_0, \mathcal{I})$.

Proof: Since the *link utilization* vector $\bar{x}$ is feasible, there exists a *stable* schedule $\mathcal{S}$ which achieves the link rates specified by $\bar{x}$. Recall the notation $X(e,t)$. Let $E_t = \{e : X(e,t) = 1\}$ denote the set of links that transmit at time t in this schedule. We now focus on any edge $e = (u,v) \in E_t$. Let $A_t(e) = E_t \cap C(e) = \{e_j = (u_j, v_j) \in C(e) : j = 1, \ldots, s\}$ be a set of links in $C(e)$ that are scheduled simultaneously at time t. Define $G_k = \{e \in \mathcal{E} : j_{min} \cdot \theta^k \leq J(e) < j_{min} \cdot \theta^k\}, \forall k \in M$. We argue below that the maximum number of links that can be simultaneously scheduled from set $Q_{t,k}(e) = A_t(e) \cap G_k$, for any edge e, at any time t and any $k \in M$ is constant. Let the links in the set $Q_{t,k}(e)$ be numbered in non-decreasing order of their lengths, so that $\ell(u_1, v_1) \leq \ell(u_2, v_2) \leq \ldots \leq \ell(u_c, v_c)$ (cf. Figure 4.4). For simultaneously successful transmission of these links, the SINR at each node v_j, and in particular at node v_c needs to be at least β.

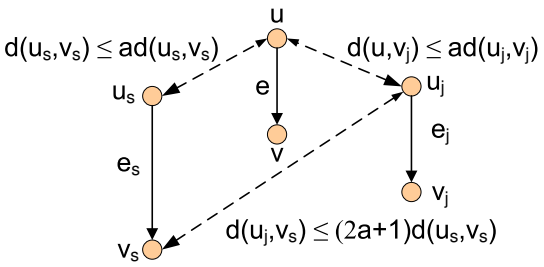


Figure 4.4: For a given link $e = (u,v)$ and set $Q_{t,k}(e)$, $d(u_j, v_c) \leq (2a+1)d(u_c, v_c)$, where $e_c, e_j \in C(e)$ and e_c is the link with longest length in set $Q_{t,k}(e)$.

Consider any $e_j, e_c \in Q_{t,k}(e), e_j \neq e_c$, we have $J(e_c)/\theta \leq J(e_j) \leq \theta J(e_c)$. Further it can be

seen that,

$$\begin{aligned} d(u_j, v_c) &\leq d(u, u_j) + d(u, u_c) + d(u_c, v_c) \\ &\leq 2ad(u, v) + d(u_c, v_c) \\ &\leq (2a+1)d(u_c, v_c), \end{aligned}$$

where the first inequality follows from the triangle inequality and the last two inequalities follow from the definition of $C(e)$, which implies that for any $e' = (u', v') \in C(e)$, we must have $d(u, u') \leq a \cdot \ell(e')$ and $\ell(e) \leq \ell(e')$.

The interference experienced at v_c due to all transmitting links in $Q_{t,k}(e) - \{e_c\}$ is

$$I_r(v_c) = \sum_{e_j=(u_j,v_j)\in Q_{t,k}(e), j\neq c} \frac{J(e_j)}{d(u_j, v_c)^\alpha}.$$

Therefore, in order to satisfy the SINR constraint at node v_c we need,

$$\frac{J(e_c)/d(u_c, v_c)^\alpha}{\left[N_0 + \sum_{e_j\in Q_{t,k}(e), j\neq c} J(e_j)/d(u_j, v_c)^\alpha\right]} \geq \beta.$$

Rearranging, we have

$$\begin{aligned} \frac{J(e_c)}{d(u_c, v_c)^\alpha} &\geq \beta \left[N_0 + \sum_{e_j\in Q_{t,k}(e), j\neq c} \frac{J(e_j)}{d(u_j, v_c)^\alpha}\right] \\ &\geq \beta N_0 + \frac{\beta J(e_c)(c-1)}{\theta(2a+1)^\alpha d(u_c, v_c)^\alpha} \\ &\geq \frac{\beta J(e_c)(c-1)}{\theta(2a+1)^\alpha d(u_c, v_c)^\alpha}. \end{aligned}$$

This in turn implies $c \leq \theta\frac{(2a+1)^\alpha}{\beta} + 1$, and therefore, we have

$$\forall e \in \mathcal{E}, \forall t, \sum_{e'\in Q_{t,k}(e)} X(e', t) \leq \eta_0.$$

Observing that $C(e) \cap H_k^i \subseteq C(e) \cap G_k$, for any T we have,

$$\forall e \in \mathcal{E}, \forall k \in M, \forall i \in L, \sum_{e'\in C(e)\cap H_k^i} \sum_{t\leq T} X(e', t) \leq T\eta_0. \tag{4.7}$$

Dividing both sides of (4.7) by T, the Lemma follows from the definition of $x(e)$ in Section 4.4.

4.6.3 Link-Flow Scheduling: sufficient conditions

The rate vector that satisfies the necessary conditions, derived in the previous Section, may result in an infeasible schedule. We derive the sufficient conditions and ensure that rate vectors satisfying these conditions yield a feasible schedule.

Scheduling Algorithm:

We assume that time is divided into sufficiently large frames, with the length of each frame $|W| = w$. Recall the definitions of Δ, Γ_{edge} and the sets H_k^i from Section 4.4. We further subdivide each frame W into $(1 + \log \Delta) \cdot (1 + \log \Gamma_{edge})$ sub-frames $W_{i,k}$, such that $|W_{i,k}| = w' = w/(1 + \log \Delta) \cdot (1 + \log \Gamma_{edge})$, $\forall i \in L, \forall k \in M$. We assume that lengths of sub-frames $W_{i,k}, \forall i \in L, k \in M$ and that $x(e) \cdot w' \forall e \in \mathcal{E}$ are integers. Algorithm FrameSchedule constructs a periodic schedule $\mathcal{S}$ by repeating a schedule $\mathcal{S}_W$ for every frame W. Within each sub-frame $W_{i,k}$, the algorithm considers only the edges from the set H_k^i and assigns $s(e) = x(e) \cdot w'$ slots for each edge $e \in H_k^i$ by a greedy coloring step. The final schedule is constructed by combining schedules $\mathcal{S}_W$ for all the frames.

For the algorithm FrameSchedule to be stable, we need to find conditions under which the algorithm correctly assigns $|s(e)| = x(e) \cdot w'$ number of slots for each $e \in \mathcal{E}$. The following Lemma proves that for a suitable choice of η, the algorithm is indeed successful.

Lemma 4.4. *Algorithm FrameSchedule correctly assigns $|s(e)| = x(e) \cdot w'$ slots for each edge e, if the* link utilization *vector $\bar{x}$ satisfies the following conditions $\forall e \in \mathcal{E}, \forall i \in L, \forall k \in M$:*

$$\sum_{e' \in C(e) \cap H_k^i} x(e') \leq \frac{1}{(1 + \log \Delta)(1 + \log \Gamma_{edge})}$$

Algorithm 1: FrameSchedule

Input : (i) $\mathcal{E}$,(ii)$\bar{x}$,(iii)W,(iv)$\bar{J}_{edge}$,(v)w

Output : Sets $s(e)$ for all edges e and schedule $\mathcal{S}_W$

for $e \in \mathcal{E}$ **do**
  $s(e) = \phi$
end
Partition W into $(1 + \log \Delta) \cdot (1 + \log \Gamma_{edge})$ sets $W_{i,k}$ of equal size, for $i \in L$, $k \in M$.
for $i = \lfloor \log \Delta \rfloor$ *downto* 0 **do**
  for $k = \lfloor \log \Gamma_{edge} \rfloor$ *downto* 0 **do**
    `//Greedy Coloring`
    Order edges in H^i_k in non-increasing order of their lengths, such that $H^i_{k,sort} = \{e_1, \ldots, e_s\}$.
    for $j = 1$ *to* $|H^i_{k,sort}|$ **do**
      $s'(e_j) = \bigcup_{e' \in C(e) \cap \{e_1, \ldots, e_{j-1}\}} s(e')$
      $s(e_j) =$ any subset of $W_{i,k} \setminus s'(e_j)$ of size $w \cdot x(e)$
    end
  end
end
Construct schedule $\mathcal{S}_W$: at each time $t \in W$, schedule all links $e \in \mathcal{E}$ with $t \in s(e)$.

This implies that $\bar{x}$ is any feasible solution to the linear program $\mathcal{P}(\frac{1}{(1+\log\Delta)(1+\log\Gamma_{edge})}, \mathcal{I})$.

Proof: Let us assume that for some edge $e_j \in H_k^i$ with *link utilization* $x(e_j)$, algorithm FrameSchedule fails to assign $s(e_j) = x(e_j) \cdot w'$ slots. Therefore, we must have,

$$\sum_{e' \in C(e_j) \cap H_k^i} |s(e')| > \frac{w}{(1+\log\Delta) \cdot (1+\log\Gamma_{edge})}.$$

Dividing both sides by w, we get $\sum_{e' \in C(e_j) \cap H_k^i} x(e') > 1/(1+\log\Delta) \cdot (1+\log\Gamma_{edge})$, which contradicts the condition on $\bar{x}$.

Next, we need to show that the Schedule produced by Algorithm 1 is indeed valid, wherein the SINR constraints at every receiver are satisfied.

Lemma 4.5. *Let $\bar{x}$ be a feasible solution to the program $\mathcal{P}(1/(1+\log\Delta)(1+\log\Gamma_{edge}), \mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge}))$. Then, Algorithm* FrameSchedule *produces a feasible schedule corresponding to $\bar{x}$ for the instance $\mathcal{I}' = (V, \mathcal{E}', \mathcal{K}, (1+\epsilon)\bar{J}_{edge}$ of* TM-SINR*, where $\mathcal{E}' = \{e \in \mathcal{E} : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$ in which the SINR constraints are satisfied at all receivers, for constants $a \geq 2\sqrt[\alpha]{\frac{48\theta\beta(1+\epsilon)}{\epsilon(\alpha-2)}}$, $\alpha > 2, \epsilon > 0$, and $\theta = 2$.*

Proof: We show that at any time t, the set E_t of links scheduled at this time in $\mathcal{S}$ can indeed be transmitted simultaneously, while satisfying the SINR constraints at each receiver.

By construction, there exists a set H_k^i such that $E_t \subseteq H_k^i$, for some $i \in L, k \in M$. Consider two edges $e_j, e_m \in E_t$ with $\ell(e_j) \leq \ell(e_m)$. Since these two edges are scheduled simultaneously, it must be the case that $e_m \notin C(e_j)$, which implies $d(u_j, u_m) > a \max\{\ell(e_j), \ell(e_m)\}$. For any $e_j \in H_k^i$, we have $J(e_j) \in [j_{min} \cdot 2^k, j_{min} \cdot 2^{k+1})$ which implies $J(e_m) \leq \theta J(e_j)$. Further, we have $\ell(e_j) \in [2^i, 2^{i+1})$, and therefore $a2^i \geq a\ell(e_j)/2$. This implies that if we place a disk of radius $a\ell(e_j)/4$ centered at the end points of each edge in E_t, all these disks would be disjoint.

Consider any $e_j = (u_j, v_j) \in E_t$. We estimate the SINR at v_j in the following manner. As in [8, 42], we partition the plane into rings R_d centered at u_j (cf. Figure 4.5) for $d = 0, 1, \ldots$, each of width $a\ell(e_j)$ around u_j. Each ring R_d consists of all links $e_m = (u_m, v_m)$, for

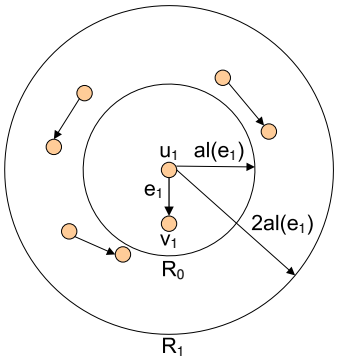


Figure 4.5: For a given link $e_j = (u_j, v_j) \in E_t$, construct rings of radius $a\ell(e_j)$ around u_j. We calculate the interference experienced by node v_j due to other simultaneously transmitting links.

which $da\ell(e_j) \leq d(u_j, u_m) < (d+1)a\ell(e_j)$. As derived earlier, for any $e_m \neq e_j$, we have $d(u_j, u_m) > a \max\{\ell(e_j), \ell(e_m)\}$, which implies R_0 does not contain any links in E_t other than e_j. The area of the ring R_d can be calculated as,

$$\begin{aligned} A(R_d) &= \pi[((d+1)a\ell(e_j))^2 - (da\ell(e_j))^2] \\ &= \pi a^2(2d+1)\ell(e_j)^2 \\ &\leq 3\pi d a^2 \ell(e_j)^2. \end{aligned}$$

and so the non-overlapping disks property implies that the number of transmitters in R_d is at most

$$\frac{3\pi d a^2 \ell(e_j)^2}{\pi a^2 \ell(e_j)^2/16} \leq 48d.$$

Next, that for each $e_m \in R_d$, we have $d(u_m, v_j) \geq (ad-1)\ell(e_j) \geq \frac{ad}{2}\ell(e_j)$, since $a > 2$. Therefore, the interference at v_j due to nodes in R_d denoted by $\mathcal{I}_d(v_j)$ is bounded as follows,

$$\begin{aligned} \mathcal{I}_d(v_j) &\leq 48 \cdot d \cdot 2^\alpha \frac{\theta \cdot J(e_j)}{(ad\ell(e_j))^\alpha} \\ &= 2^\alpha \frac{48 \cdot \theta \cdot J(e_j)}{a^\alpha d^{\alpha-1} \ell(e_j)^\alpha}. \end{aligned}$$

Summing up the interference over all rings R_d, we have,

$$\begin{aligned}\sum_{d=1}^{\infty}\mathcal{I}_d(v_j) &\leq 2^{\alpha}\frac{48\cdot\theta\cdot J(e_j)}{a^{\alpha}\ell(e_j)^{\alpha}}\sum_{d=1}^{\infty}\frac{1}{d^{\alpha-1}}\\ &\leq 2^{\alpha}\frac{48\cdot\theta\cdot J(e_j)}{a^{\alpha}\ell(e_j)^{\alpha}}\int_1^{\infty}\frac{dx}{x^{\alpha-1}}\\ &\leq \frac{2^{\alpha}48\cdot\theta\cdot J(e_j)}{a^{\alpha}\ell(e_j)^{\alpha}(\alpha-2)}.\end{aligned}$$

Therefore the SINR at receiver v_j is at least

$$\begin{aligned}SINR(v_j) &\geq \frac{J(e_j)}{\ell(e_j)^{\alpha}[N_0+\frac{2^{\alpha}48\cdot\theta\cdot J(e_j)}{a^{\alpha}\ell(e_j)^{\alpha}(\alpha-2)}]}\\ &= \frac{J(e_j)}{\ell(e_j)^{\alpha}[N_0+\frac{\epsilon J(e_j)}{(1+\epsilon)\beta\ell(e_j)^{\alpha}}]},\end{aligned}$$

which is at least β if $a^{\alpha}\geq 2^{\alpha}\frac{48\theta\beta(\epsilon+1)}{\epsilon(\alpha-2)}$ and $J(e_j)\geq(1+\epsilon)\beta N_0\ell(e_j)^{\alpha}$. We therefore choose a that satisfies the above condition. In order guarantee the feasibility and validity of the schedule $\mathcal{S}$, we add a small slack ϵ to the power levels and enforce the following constraints: $\forall e\in\mathcal{E}, J(e)\geq(1+\epsilon)\beta N_0\ell(e)^{\alpha}$. This corresponds to a *different* instance $\mathcal{I}'=(V,\mathcal{E}',\mathcal{K},(1+\epsilon)\bar{J}_{edge})$ of TM-SINR, where $\mathcal{E}'=\{e\in\mathcal{E}: J(e)\geq(1+\epsilon)\beta N_0\ell(e)^{\alpha}\}$. The vector $\bar{x}$ produced by algorithm FrameSchedule is therefore not valid for the original instance $\mathcal{I}=(V,\mathcal{E},\mathcal{K},\bar{J}_{edge})$, but is valid for the instance $\mathcal{I}'$ of TM-SINR.

It should be noted that, in order guarantee the feasibility and validity of the schedule $\mathcal{S}$, we enforce the following constraints on the input power levels: $\forall e\in\mathcal{E}, J(e)\geq(1+\epsilon)\beta N_0\ell(e)^{\alpha}$. This corresponds to a *different* instance $\mathcal{I}'=(V,\mathcal{E}',\mathcal{K},(1+\epsilon)\bar{J}_{edge})$ of TM-SINR, where $\mathcal{E}'=\{e\in\mathcal{E}: J(e)\geq(1+\epsilon)\beta N_0\ell(e)^{\alpha}\}$. The vector $\bar{x}$ produced by algorithm FrameSchedule is therefore not valid for the original instance $\mathcal{I}=(V,\mathcal{E},\mathcal{K},\bar{J}_{edge})$, but is valid for the instance $\mathcal{I}'$ of TM-SINR.

4.6.4 Putting everything together

For an input instance $\mathcal{I}'=(V,\mathcal{E}',\mathcal{K},(1+\epsilon)\bar{J}_{edge})$ of TM-SINR, we first compute the optimum *link utilization* vector $\bar{x}$ by solving the linear program $\mathcal{P}(1/(1+\log\Delta)\cdot(1+\log\Gamma_{edge}),\mathcal{I}=$

$(V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$, where $\mathcal{E}' = \{e \in \mathcal{E} : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$. We know from Lemma 4.5 that $\bar{x}$ can be scheduled feasibly for the instance $\mathcal{I}'$. The following theorem shows that the rate achieved by $\bar{x}$ is within a provable factor of $r_{opt}(\mathcal{I})$ - thus, this is a **bi-criteria** approximation, in which we compare the quality of the solution produced by our algorithm with respect to the optimum for an instance that uses slightly less power.

Theorem 4.1. *Let $\mathcal{I}' = (V, \mathcal{E}', \mathcal{K}, (1+\epsilon)\bar{J}_{edge})$ be an instance of* TM-SINR*, and let $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$ be the corresponding instance for which the optimum rate $r_{opt}(\mathcal{I})$ is considered, such that $\mathcal{E}' = \{e \in \mathcal{E} : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$, for any $\epsilon > 0$. The optimum solution $\bar{x}$ to the program $\mathcal{P}(1/(1+\log\Delta)\cdot(1+\log\Gamma_{edge}), \mathcal{I})$ yields a feasible and stable* link utilization *vector for the instance $\mathcal{I}'$, and results in a total throughput of at least $\Omega(r_{opt}(\mathcal{I})/\eta_0(1+\log\Delta)\cdot(1+\log\Gamma_{edge}))$, for $\eta_0 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2$, and a as defined in Chapter 2, Section 2.5.4.*

Proof: Let $\bar{x}_{opt}$ be the optimum utilization vector for the instance $\mathcal{I}$ of TM-SINR, achieving a total throughput rate of $r_{opt}(\mathcal{I})$. From Lemma 4.3, it follows that $\bar{x}_{opt}$ is a feasible solution if it satisfies the conditions stated in Lemma 4.3 and hence is a feasible solution to the program $\mathcal{P}(\eta_0, \mathcal{I})$, for $\eta_0 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2$, and a as defined in Chapter 2, Section 2.5.4. We now scale down the *link utilization* vector $\bar{x}_{opt}$ to achieve a new vector $\bar{y}$ such that $\bar{y} = \frac{1}{\eta_0(1+\log\Delta)\cdot(1+\log\Gamma_{edge})}\bar{x}_{opt}$. Since $\mathcal{P}(\eta_0, \mathcal{I})$ is a linear program, it follows that $\bar{y}$ is a feasible solution to the program $\mathcal{P}(1/(1+\log\Delta)\cdot(1+\log\Gamma_{edge}), \mathcal{I})$, and results in a total throughput rate of $\frac{r_{opt}(\mathcal{I})}{\eta_0(1+\log\Delta)\cdot(1+\log\Gamma_{edge})}$. This implies that the optimum solution $\bar{x}$ to the program $\mathcal{P}(1/(1+\log\Delta)\cdot(1+\log\Gamma_{edge}), \mathcal{I})$ also results in a total throughput rate of at least $\frac{r_{opt}(\mathcal{I})}{\eta_0(1+\log\Delta)\cdot(1+\log\Gamma_{edge})}$. Finally, by Lemma 4.5, it follows that $\bar{x}$ can be scheduled feasibly for the instance $\mathcal{I}'$ of TM-SINR. Therefore, the theorem follows.

4.7 TM-SINR for Uniform Power Levels

In the previous Section, we considered a generic case of power levels, where in every edge $e \in \mathcal{E}$ had an assigned power level $J(e)$. We now consider a specific case of power levels in which all edges $e \in \mathcal{E}$ have the *uniform* (same) power level $J(e) = J$, where $J \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha, \forall e \in \mathcal{E}$. We show that the approximation bound of $O((1+\log\Delta)\cdot(1+\log\Gamma_{edge}))$ derived on the achievable throughput (cf. Theorem 4.1) can be improved to a $(1+\log\Delta)$ approximation for the case of *uniform* power levels.

4.7.1 Problem Formulation

We consider input instances of TM-SINR specified as $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, J)$, with *uniform* power level of $J(e) = J$ for every edge $e \in \mathcal{E}$. The problem formulation for the TM-SINR problem for *uniform* power levels is similar to the one presented in Section 4.6.1. Recall that in Section 4.6.1, we partitioned the set $\mathcal{E}$ of edges into sets H_k^i based on the edge lengths and power levels, (i.e. $H_k^i = \{e = (u,v) \in \mathcal{E} : \ell(e) \in [2^i, 2^{i+1}) \wedge J(e) \in [j_{min}\cdot 2^k, j_{min}\cdot 2^{k+1})\}, \forall i \in L, k \in M$). Since the power levels are *uniform*, we only need to partition the set $\mathcal{E}$ of edges based on edge lengths. Therefore we obtain sets $H^i = \{e = (u,v) \in \mathcal{E} : \ell(e) \in [2^i, 2^{i+1})\}, \forall i \in L$. For an instance $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, J_e)$ of TM-SINR, we define a different formulation $\mathcal{P}_u(\eta, \mathcal{I})$ by replacing the constraints (4.5) in the program $\mathcal{P}(\eta, \mathcal{I})$ by the constraints

$$\forall e \in \mathcal{E}, \forall i \in L \sum_{e' \in C(e) \cap H^i} x(e') \leq \eta. \tag{4.8}$$

4.7.2 Link-Flow Scheduling: Necessary Conditions

Lemma 4.6. *Let $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, J_e)$ be an instance of the* TM-SINR *problem with* uniform *power level J, and let $\bar{x} \in \mathcal{X}(\mathcal{I})$ be any feasible* link utilization *vector. Then, $\bar{x}$ satisfies the*

following conditions:

$$\forall e \in \mathcal{E}, \forall i \in L, \sum_{e' \in C(e) \cap H^i} x(e') \leq \eta_1,$$

where $\eta_1 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 1$, *and* a *is the constant defined in Chapter 2, Section 2.5.4. This implies that* $\bar{x}$ *is a feasible solution to the program* $\mathcal{P}_u(\eta_1, \mathcal{I})$.

Proof: The proof is similar to that of Lemma 4.3. Recall the notations used in the proof of Lemma 4.3. For the case of uniform power levels, we only consider the sets $H^i \forall i \in L$. We further do not consider the set G_k and set $Q_t(e) = A_t(e)$. By following the sequence of steps used in Lemma 4.3 and substituting $\theta = 1$ the Lemma follows.

4.7.3 Link-Flow Scheduling: Sufficient Conditions

We now consider the sufficient conditions for link-flow stability for the case of *uniform* power levels. Algorithm UniformFrameSchedule is the modified scheduling algorithm for this setting. As in algorithm 1, we assume that time is divided into sufficiently large frames (W) of length w. We subdivide each frame W into $(1 + \log \Delta)$ sub-frames W_i each of length $w' = w/(1 + \log \Delta) \forall i \in L$. We assume that w' and $x(e) \cdot w' \forall e \in \mathcal{E}$ are integrals. Algorithm UniformFrameSchedule constructs a periodic schedule $\mathcal{S}$ by repeating a schedule $\mathcal{S}_W$ for every frame W. Within each sub-frame W_i, the algorithm considers only the edges from the set H^i and assigns $s(e) = x(e) \cdot w'$ slots for each edge $e \in H^i$ by a greedy coloring step.

We construct a periodic schedule $\mathcal{S}$ using Algorithm 2 by repeating the schedule $\mathcal{S}_W$ for each frame W.

Lemma 4.7. *Algorithm* UniformFrameSchedule *correctly assigns* $|s(e)| = x(e) \cdot w'$ *slots for each edge* e, *if the* link utilization *vector* $\bar{x}$ *is any feasible solution to the program* $\mathcal{P}(\frac{1}{(1+\log \Delta)}, \mathcal{I})$.

Proof: The proof is similar to that of Lemma 4.4. Since we have *uniform* power levels,

Algorithm 2: UniformFrameSchedule

Input : (i) $\mathcal{E}$, (ii) $\bar{x}$, (iii) W, (iv) w

Output : Set $s(e)$ for all $e \in \mathcal{E}$, and schedule $\mathcal{S}_W$

for $e \in \mathcal{E}$ **do**
  $s(e) = \phi$
end
Partition W into $1 + (\log \Delta)$ sets W_i of equal size, for $i \in L$,
for $i = \lfloor \log \Delta \rfloor$ *downto* 0 **do**
  `//Greedy Coloring`
  Order edges in H^i in non-increasing order of their lengths to obtain $H^i_{sort} = \{e_1, \ldots, e_s\}$.
  for $j = 1$ *to* $|H^i_{sort}|$ **do**
    $s'(e_j) = \bigcup_{e' \in C(e) \cap \{e_1, \ldots, e_{j-1}\}} s(e')$
    $s(e_j) =$ any subset of $W_i \setminus s'(e_j)$ of size $x(e) \cdot w$
  end
end
Construct schedule $\mathcal{S}_W$: at each time $t \in W$, schedule all links $e \in \mathcal{E}$ with $t \in s(e)$.

$\log \Gamma_{edge} = 0$. Further we only consider sets $H^i, \forall i \in L$. By making these modifications to Lemma 4.4, it follows that algorithm UniformFrameSchedule indeed assigns $x(e) \cdot w'$ slots for each edge e. We now derive the conditions under which the schedule is valid.

Lemma 4.8. *Let $\bar{x}$ be a feasible solution to the program $\mathcal{P}_u(1/(1+\log \Delta), \mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge}))$, Then, Algorithm* UniformFrameSchedule *produces a feasible schedule corresponding to $\bar{x}$ for the instance $\mathcal{I}' = (V, \mathcal{E}', \mathcal{K}, (1+\epsilon)\bar{J}_{edge}$ of* TM-SINR*, where $\mathcal{E}' = \{e \in \mathcal{E} : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$ in which the SINR constraints are satisfied at all receivers, for constants $a \geq 2\sqrt[\alpha]{\frac{48\theta\beta(1+\epsilon)}{\epsilon(\alpha-2)}}$, $\alpha > 2, \epsilon > 0$, and $\theta = 1$.*

Proof: By considering only sets $H^i, \forall i \in L$ and by substituting $\log \Gamma_{edge} = 0, \theta = 1$, in the proof of Lemma 4.5, we can prove the above Lemma.

Theorem 4.2. *Let $\mathcal{I}' = (V, \mathcal{E}', \mathcal{K}, (1+\epsilon)\bar{J}_{edge})$ be an instance of* TM-SINR*, and let $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$ be the corresponding instance for which the optimum rate $r_{opt}(\mathcal{I})$ is considered, such that $\mathcal{E}' = \{e \in \mathcal{E} : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$, for any $\epsilon > 0$. The optimum solution $\bar{x}$ to the program $\mathcal{P}_u(1/(1+\log\Delta), \mathcal{I})$ yields a feasible and stable* link utilization *vector for the instance $\mathcal{I}'$, and results in a total throughput of at least $\Omega(r_{opt}(\mathcal{I})/\eta_1(1+\log\Delta))$, for $\eta_1 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 1$, and a as defined in Chapter 2, Section 2.5.4.*

Proof: The proof of Theorem 4.1 can be applied here by substituting $\log\Gamma_{edge} = 0$, η_0 with η_1 and program $\mathcal{P}$ with $\mathcal{P}_u$.

4.8 TM-SINR for Linear Power Levels

We now consider another special case of power levels, in which $J(e) = c_1\ell(e)^\alpha, \forall e \in \mathcal{E}$ for constant c_1 such that $c_1 \geq (1+\epsilon)\beta N_0$ - this is also called the *linear* power level. Theorem 4.1 implies an approximation of $O((1+\log\Delta)^2)$ for this case, since $\log\Gamma_{edge} = O(\log\Delta)$. In this Section, we show that this bound can be improved to $O(1+\log\Delta)$.

Let $\bar{J}_{edge}$ be the power value vector with $J(e) = c_1\ell(e)^\alpha, \forall e \in \mathcal{E}$. Recall the definition of the sets H^i from Section 4.7.1. In order to get a better approximation, we partition the set of edges $\mathcal{E}$ into sets H^i based on their lengths. It can be seen that $\forall e', e'' \in H^i, J(e')/2^\alpha \leq J(e'') \leq J(e')$, $\forall i \in L$. For an instance $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$ of TM-SINR, we consider the program $\mathcal{P}_u(\eta, \mathcal{I})$ (described in Section 4.7.1) instead of the program $\mathcal{P}(\eta, \mathcal{I})$.

Lemma 4.9. *Let $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$ be an instance of the* TM-SINR *problem with* linear *power levels $\bar{J}_{edge}$, and let $\bar{x} \in \mathcal{X}(\mathcal{I})$ be any feasible* link utilization *vector. Then, $\bar{x}$ satisfies the following conditions:*

$$\forall e \in \mathcal{E}, \forall i \in L, \sum_{e' \in C(e) \cap H^i} x(e') \leq \eta_2,$$

where $\eta_2 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2^\alpha$, *and* a *is the constant defined in Chapter 2. This implies that* $barx$ *is a feasible solution to the program* $\mathcal{P}_u(\eta_2, \mathcal{I})$.

Proof: The proof of Lemma 4.6 can be applied here, by substituting $\theta = 2^\alpha$.

Theorem 4.3. *Let* $\mathcal{I}' = (V, \mathcal{E}', \mathcal{K}, (1+\epsilon)\bar{J}_{edge})$ *be an instance of* TM-SINR*, and let* $\mathcal{I} = (V, \mathcal{E}, \mathcal{K}, \bar{J}_{edge})$ *be the corresponding instance for which the optimum rate* $r_{opt}(\mathcal{I})$ *is considered, such that* $\mathcal{E}' = \{e \in \mathcal{E} : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$*, for any* $\epsilon > 0$*. The optimum solution* $\bar{x}$ *to the program* $\mathcal{P}_u(1/(1+\log\Delta), \mathcal{I})$ *yields a feasible and stable* link utilization *vector for the instance* $\mathcal{I}'$*, and results in a total throughput of at least* $\Omega(r_{opt}(\mathcal{I})/\eta_2(1+\log\Delta))$*, for* $\eta_2 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2^\alpha$*, and* a *as defined in Chapter 2, Section 2.5.4.*

Proof: The proof of is similar to that of Lemmas 4.7,4.8, and Theorem 4.2. We use algorithm UniformFrameSchedule to schedule the vector $\bar{x}$. By substituting $\theta = 2^\alpha$ and η_1 with η_2 in the proof of Lemma 4.8 and Theorem 4.2, the theorem follows.

4.9 Most Relevant Prior Work

Our work with the TM-SINR problem is most closely related to Kumar *et al.* [30]. In [30], the authors provide a constant approximation algorithm for the throughput maximization problem along with joint scheduling and routing for a single channel, single radio network. The results provided in [30] are based on a linear programming (LP) formulation that captures the properties of wireless interference models. The results hold for various interference models such as protocol, transmitter, transmitter-receiver model. The scheduling framework provided in [30] however cannot be easily extended to incorporate the SINR interference model. In our work, we exploit the geometric nature of the SINR model to develop a different scheduling framework. We combine some of the techniques discussed in [30] such as

linear relaxation to derive the bounds on the network capacity.

4.10 Summary

We studied the problem of maximizing throughput of any given wireless network with SINR constraints. By extending the recent work of [8, 30], we developed the first provable algorithms for this problem by means of a linear programming formulation. We constructed instances and compared the throughput achieved by the SINR and disk-graph interference model. We demonstrate that these two models can provide throughput predictions that significantly differ from each other suggesting the need for greater care in using these models.

Chapter 5

Experimental Evaluation for the TM-SINR problem

In the previous chapter we studied the throughput maximization problem with the SINR interference model (TM-SINR) for any given instance of a wireless network. We developed an approximation algorithm that has flavors of cross-layer design, and theoretically derived upper and lower bounds on the network capacity. In this chapter we conduct extensive simulations to validate the approximation techniques described in Chapter 4, gain deeper insights into the theoretical model, and understand the impact of cross-layer design on the network capacity. Specifically, the goals of our simulations are as follows:

1. *Validation of the theoretical model*: In this set of experiments, we verify if the schedule produced by the greedy algorithm FrameSchedule (cf. Chapter 4, Section 4.6.3) is feasible (i.e. SINR constraints are not violated at any given time) by implementing it in a realistic network simulator. We observe that the greedy scheduling scheme is feasible and the rates obtained by the linear program (LP) for every connection (cf. Chapter 4, Section 4.6) can be realized in a realistic setting.

2. *Approximate vs. optimal solution*: For a reasonable network setting, we compute the

optimal solution, compare it with the approximate solution and compute the approximation ratio. We study the impact of different network types on the performance of the approximation algorithm. We observe that the theoretical approximation bound derived by our technique is indeed a worst case bound and in practice our model can obtain higher performance gains. We identify conditions under which the performance of the approximation algorithm can be limited and recognize certain parameters that can allow us to improve the performance of our model.

3. *Impact of fairness*: In this set of experiments, we first study the unconstrained network throughput as determined by the approximation algorithm and study the impact of fairness constraints on it (cf. Chapter 4, Section 4.6). We observe that the overall system throughput goes down as the system becomes more fair, implying that complete starvation of certain flows is possible when there is very little or no fairness in the system.

4. *Impact of Cross-layer interaction*: We implement a congestion-aware path selection heuristic based on the congestion metric discussed in Chapter 2. This heuristic attempts to select paths that have low MAC-level congestion. We observe that this routing scheme performs better than a hop-count based shortest path routing scheme, thereby implying that interaction between the MAC and the routing layer can assist in improving the performance of the system.

5.1 Simulation Setup

In this section, we describe the common simulation set up used for all experiments. Table 5.1 summarizes the parameters used in the experiments.

Network types:

- Geometric Random Network: We consider a geometric random network formed by

uniform random distribution of 225 nodes in a $50m \times 50m$ square units area.

- Grid Network: We consider a grid network of 15×15 nodes, with uniform grid spacing of $10m$.

- Realistic Network: We consider a road traffic network corresponding to a distribution of 227 cars for a particular time instance in a region of downtown Portland, OR, obtained by running the TRANSIMS simulator [56]. We scaled down this network to fit in a $50m \times 50m$ region.

Path-loss model:

We consider the Regression path-loss model [45], which is a generic version of the Free-space path-loss model. For a given transmission link, $e = (u, v)$, with u as the transmitter and v as the receiver, the received power $P_r(v)$ under this model can be expressed as

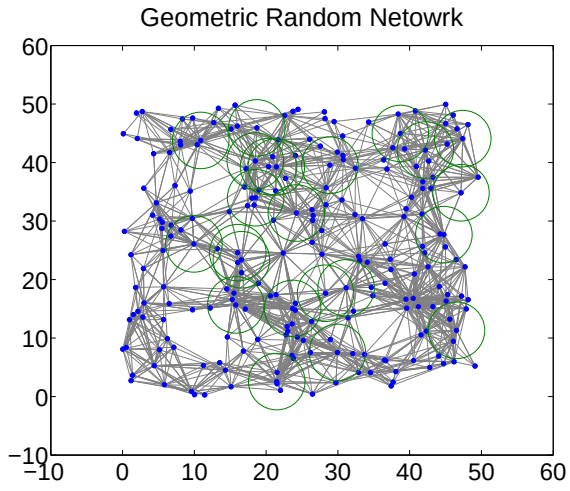
$$P_r = \frac{P_t(u)}{d(u,v)^\alpha} \frac{\lambda}{4\pi}$$

where $P_t(u)$ denotes the transmission power, $d(u, v)$ is the Euclidean distance between the two nodes, α denotes the path-loss exponent, and λ is the wavelength. This model assumes that the received power does not obey a simple inverse square law (as assumed in the Free-space model), but rather depends on the distance according to $1/d^\alpha$. The effects of various propagation schemes are embedded within the value of α, which can lie between the range of 2,6 [42].

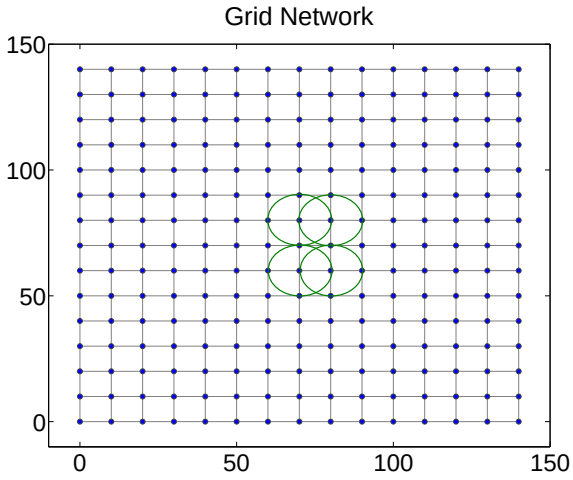
Other Settings:

- Number of connections: We experiment with varying number of connections, each of the source-destination pairs are chosen u.a.r. We denote these as k.

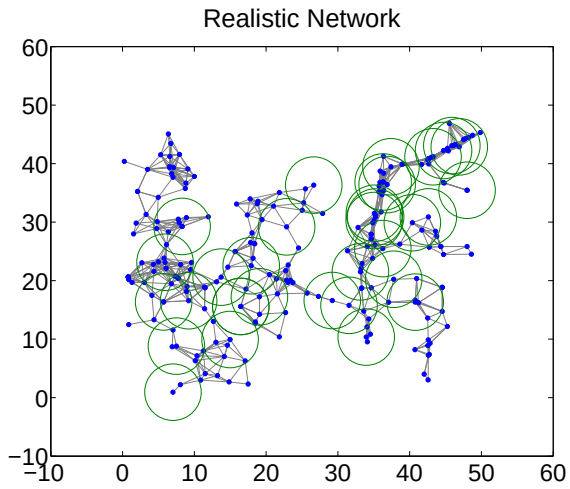
- Edge Capacities: All edges have a transmission rate of 5Mbps.



(a) Geometric Random Network



(b) Grid Network



(c) Realistic Network

Figure 5.1: (a) Geometric Random Network consisting of 225 nodes in a $50m \times 50m$ square area. (b) Grid network consisting of 225 nodes with uniform spacing of $10m$. (c) Realistic Network consisting of 227 nodes in a $50m \times 50m$ square area.

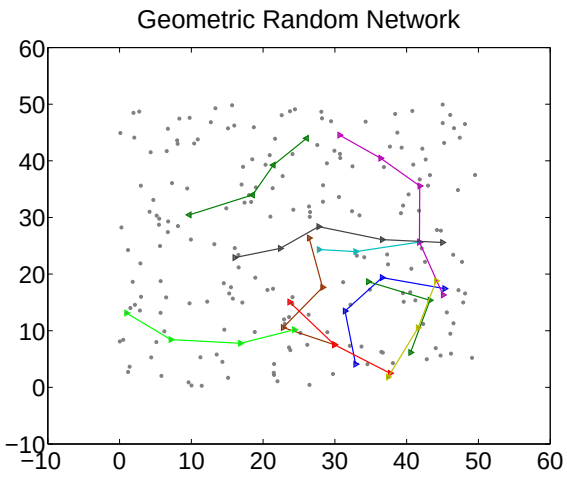
- Transmission power: We perform all experiments for the case of uniform power levels, where every transmitter can transmit at 40mW. The transmission range was set to 10m.
- Number of seeds: All data points are averaged over 5 runs of the experiment.

Physical interference model	SINR
Physical model	802.11
Network Protocol	Internet Protocol (IP)
Transport protocol	UDP
Queue Size	12500000 bytes
Propagation Path-loss Model	Regression
Number of time slots per frame	200
Slot Duration	2.470 mili sec
Number of packets	2000
Packet Size	1500 bytes
α	6
ϵ	8
β	$1mW$
a	3.08
Channel Frequency	2.4 GHz
Channel Bandwidth	2MHz
Transmission Power	40 mW
Transmission Range	10 m
System loss	4.9 mW
Ambient Noise	$4 \times 10^{-19} mW/Hz$
Simulation Time	800 secs
Network Simulator	Qualnet [47]
LP Solver	Neos [52]

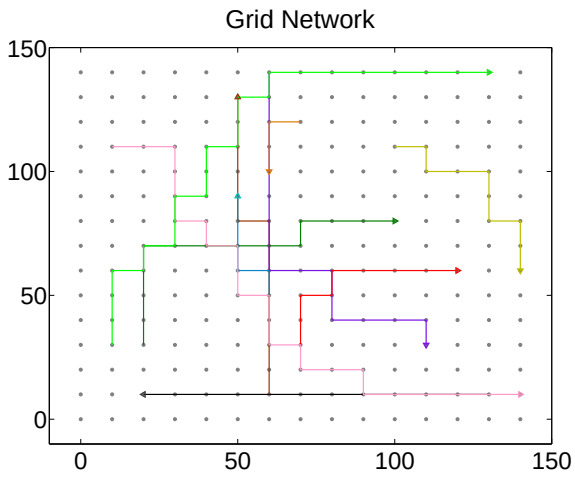
Table 5.1: Parameters used in the experiments

5.2 Validation of the theoretical model

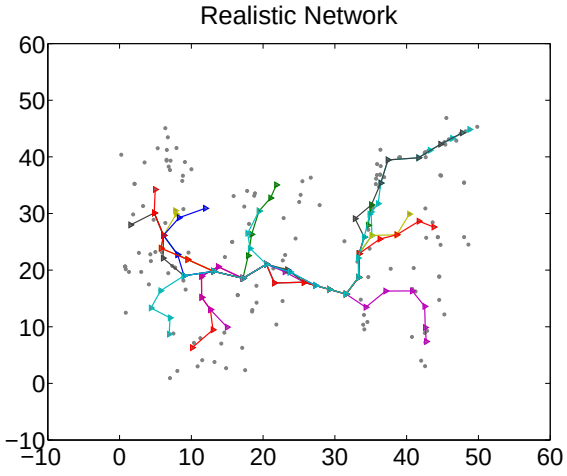
In Chapter 4, we formulated a linear programming (LP) to approximate the maximum throughput rate vector (cf. Section 4.6). We then developed a centralized greedy scheduling



(a)



(b)



(c)

Figure 5.2: Source-destination pairs are chosen u.a.r and shortest path is computed between every pair. (a) Geometric Random Network consisting of 225 nodes in a $50m \times 50m$ square area. (b) Grid Network consisting of 225 nodes, with uniform grid spacing of $10m$ units. (c) Realistic Network consisting of 227 nodes in a $50m \times 50m$ square area.

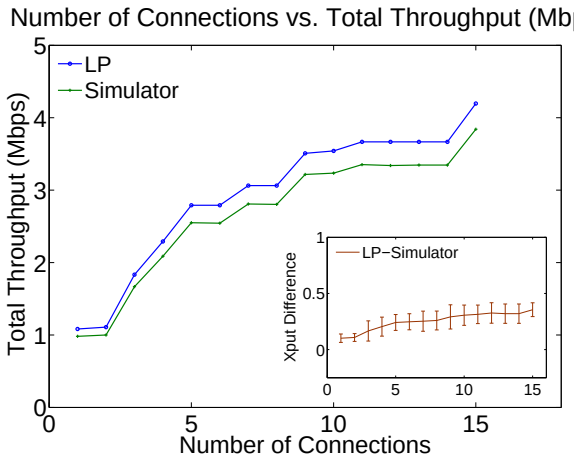
scheme (FrameSchedule cf. Section 4.6.3) and theoretically proved that the rate vectors obtained by solving the LP can be feasibly scheduled without violating the SINR constraints (Section 4.6.3). In this set of experiments, we verify the correctness of our theoretical model.

Goal: The goal of this experiment is to verify the feasibility of the rates and schedule derived by the approximation algorithm for the TM-SINR problem in a realistic setting.

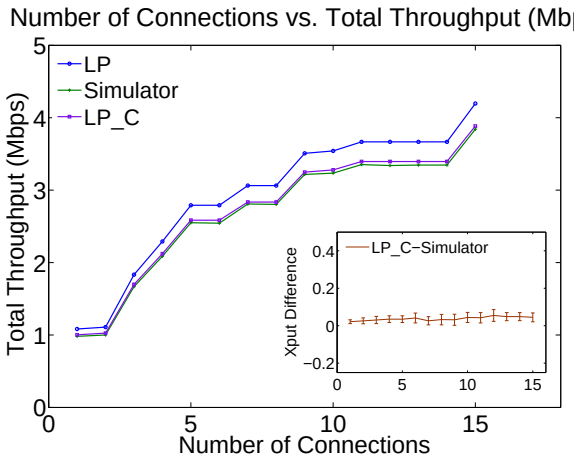
For a given network instance, we solve the LP using the Neos solver [52] and obtain the overall throughput and individual link rates. We then construct a centralized TDMA schedule using the greedy scheduling algorithm FrameSchedule and implement this schedule in the Qualnet simulator. The simulator decides if a packet has been successfully received or not by measuring the signal-to-noise ratio at every receiver and comparing it with the SINR threshold (β). We observe the overall throughput achieved by the simulator for a given schedule and compare this with the throughput obtained by solving the LP. We study the variation of throughput as a function of the number of connections. In this set of experiments, we do not consider the impact of routing; for a given set of randomly chosen source destination pair, we compute the shortest path using the Dijkstra's algorithm. The shortest path routes chosen for a particular seed are shown in Figure 5.2. We do not consider the impact of fairness constraints and set the fairness index $\lambda = 0$. We repeat this experiment for different network types.

Hypothesis: We expect the throughput obtained by the LP to match closely with the throughput obtained by the simulator.

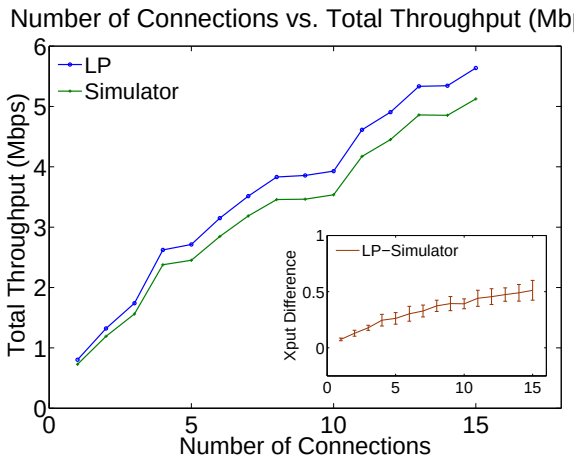
Results and Explanation: Figures 5.3(a),(b),(c),(d) summarize the results for different network types averaged over 5 runs. In every figure, we also plot the average difference between the LP and the simulator output for different number of connections and provide 95% confidence intervals according to the Gaussian distribution. We observe differences between the LP and the simulator output. These are mainly due to the delay introduced by transmission of control packets in the simulator. At the physical layer of the simulator, according



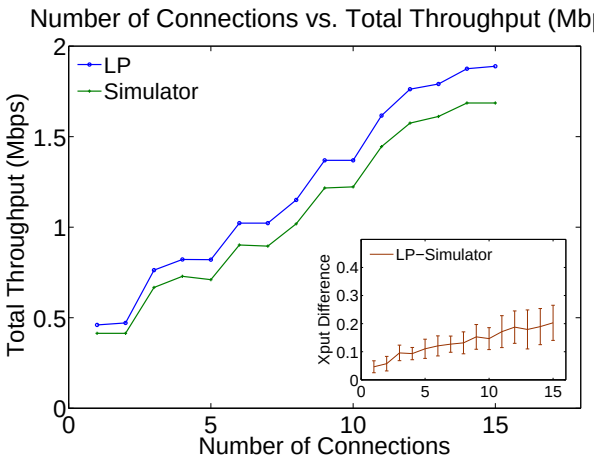
(a) Geometric Random Network



(b) Geometric Random Network



(c) Grid Network



(d) Realistic Network

Figure 5.3: Variation of throughput w.r.t. the number of connections as observed by the LP and the network simulator for three different network types. There are no fairness constraints.

to the 802.11 specification, a control packet (known as the PLCP preamble) is sent before transmitting a data packet. This packet has a fixed size of 192 bits, is sent at a rate of 1Mbps, and is responsible for synchronization (clock recovery) between the sender and the receiver (we refer interested readers to the 802.11 specifications for more details [14]). In the greedy scheduling algorithm (FrameSchedule), we divide the entire time frame into time-slots of equal lengths. In order to ensure that the packets sent at the start of every time-slot reach before or at the end of every time-slot, we set the duration of the time-slot to be an integral multiple of the transmission time. For example, if the transmission rate of every edge is 5Mbps, and the packet size is 1000 bytes, the transmission time is $1000 * 8/5 = 1600\mu$secs. The slot duration in this example would by $k \cdot 1600\mu secs$, where k is the number of packets sent in a given time-slot. The throughput in the scheduling algorithm is measured at the end of every time slot and is calculated as *total number of bytes received at the end of the time-slot/duration of the time-slot.* We do not consider the effects of propagation delay and the delay due to the transmission of control packets (control-delay). These delays are considered in the simulator. For successful transmission and reception of packets in any given time-slot, the simulator requires the duration of the time-slot to be slightly higher than that assumed in the greedy scheduling scheme. This causes the simulator throughput to be slightly different than the LP throughput. However, we can incorporate the effect of the control packets and the propagation delay in the LP formulation. The maximum link capacity of every link in the LP can be calculated as *packet size*$\cdot 8/$*transmission time + propagation delay + control delay.* We call this LP as $LP_{compensated}$ and it can be seen from Figure 5.3(b), that the throughput achieved by $LP_{compensated}$ (denoted as LP_C in the plot) matches very closely with the simulator throughput. This shows that packet loss does not occur in the simulator due to the violations of the SINR constraints.

Conclusion: We conclude that:

1. The greedy scheduling algorithm FrameSchedule is feasible

2. The rates obtained by the LP are achievable in a realistic setting

3. Effects of various delays such as propagation, control etc. can be incorporated in the LP

5.3 Comparison with the optimal solution

In the approximation technique discussed in Chapter 4, recall that we utilize a linear relaxation scheme, and derive the necessary and sufficient conditions for the feasible rate region (cf. Sections 4.6.2, 4.6.3). These conditions correspond to the upper and lower bounds on network capacity. Based on these bounds, we derive an approximation ratio that indicates how far the approximate solution can be from the optimal solution in the worst case. In this set of experiments, we aim to gain deeper insights into our approximation techniques and the bounds by performing a comparison with the optimal solution.

Goal: The goals of this experiment are to (a) observe how far the approximate solution is from the optimal solution for a reasonable network setting, (b) compare observed approximation ratio with theoretically estimated approximation ratio, (c) study the impact of different network types on performance of the approximation technique, and (d) identify parameters for improving the performance of the theoretical model.

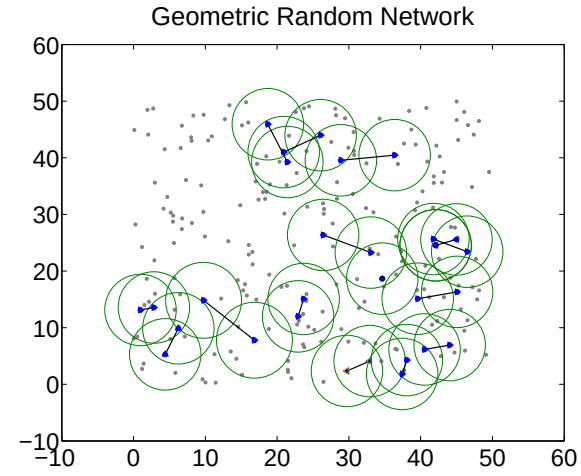
For a given network instance, we solve the LP with the sufficiency conditions and calculate the optimal solution by performing an exhaustive search (brute-force technique). Since the brute-force technique is unlikely to scale for networks with multi-hop paths, we consider only one-hop source destination pairs. We also run the LP with the necessary conditions, in order to obtain an upper bound on the throughput. For a given network instance, we compute the *observed approximation ratio*, which is the ratio of the optimal solution and the LP solution and compare this with the theoretically derived worst-case approxima-

tion ratio, which is the ratio of the derived upper bound and the lower bound (cf. Section 4.6.4). For the case of uniform power levels and uniform link capacities this ratio is $\min\{k \cdot cap, \frac{(2a+1)^{\alpha}+\beta)(1+\log\Delta)}{\beta}\}$, where k denotes the number of connections, cap denotes the maximum transmission rate of every link and a, α, β, Δ are as defined in Chapter 2. We further compare the LP throughput with the one obtained by running the standard 802.11 random-access MAC protocol in the network simulator. We study the variation of overall throughput as a function of the number of connections. We repeat this experiment for different network types. We do not consider the impact of fairness constraints on the LP and set the fairness index to 0.

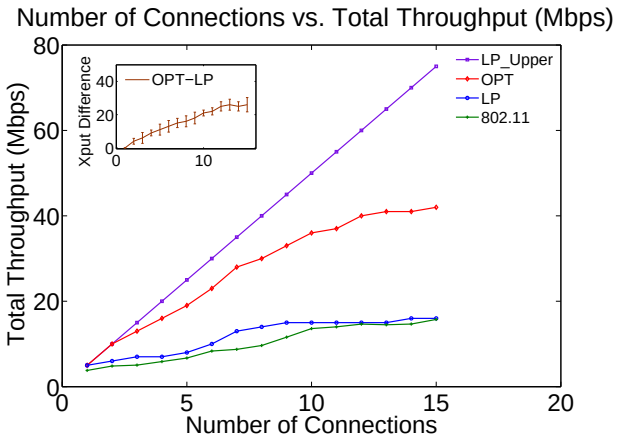
# connections	OR-grid	OR-random	OR-realistic	TR
1	1	1	1	1
2	1	1.66667	1.11111	10
3	1	1.85714	1.27273	15
4	1.05263	2.28571	1.46154	20
5	1.04167	2.375	1.84615	25
6	1.16	2.3	1.93333	30
7	1.30769	2.15385	1.78947	35
8	1.31034	2.14286	1.94737	40
9	1.43333	2.2	2.10526	45
10	1.46875	2.4	2.1	50
11	1.625	2.46667	2.19048	55
12	1.64706	2.66667	2.38095	60
13	1.74286	2.73333	2.40909	65
14	1.75	2.5625	2.43478	70
15	1.94286	2.625	2.5	75

Table 5.2: Comparison of the observed approximation ratio with the theoretically derived approximation ratio for the different network types. OR-grid, OR-rand, OR-realistic denotes the ratio of the optimal solution to the LP solution for the grid, random and realistic network respectively. TR denotes the ratio of the upper bound and the lower bound

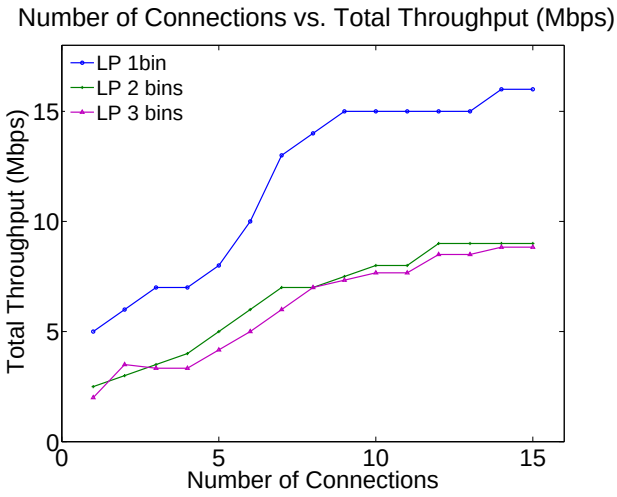
Results and Explanation: Figures 5.4, 5.6, 5.7 plot the results of this experiment for different network types. We classify our observations in the following way:



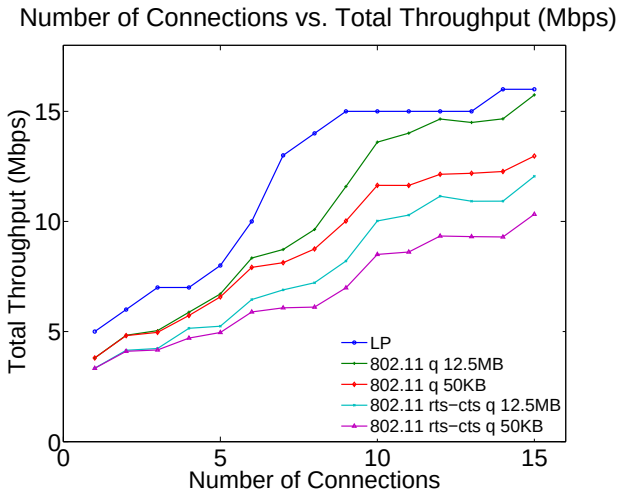
(a)



(b)



(c)



(d)

Figure 5.4: Simulation results for a geometric random network consisting of 225 nodes in a $50m \times 50m$ square area. There are no fairness constraints. (a) Instance of the random network for a particular seed. One-hop source-destination pairs are selected u.a.r. (b) Variation of throughput w.r.t. the number of connections as observed by the optimal solution, LP and 802.11 protocol. LP_Upper denotes the upper bound computed by running the LP with the necessary conditions. The 802.11 protocol ran without rts-cts and the queue size was fixed to 12.5MB. The total number of bins in the LP was set to one. (c) Different binning strategies were considered for the LP. (d) 802.11 protocol was considered with and without rts-cts messaging and for two different queue sizes, 12.5Mbytes and 50Kbytes respectively.

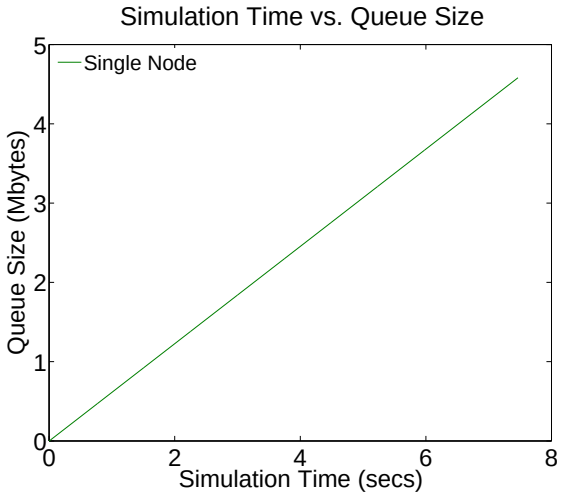
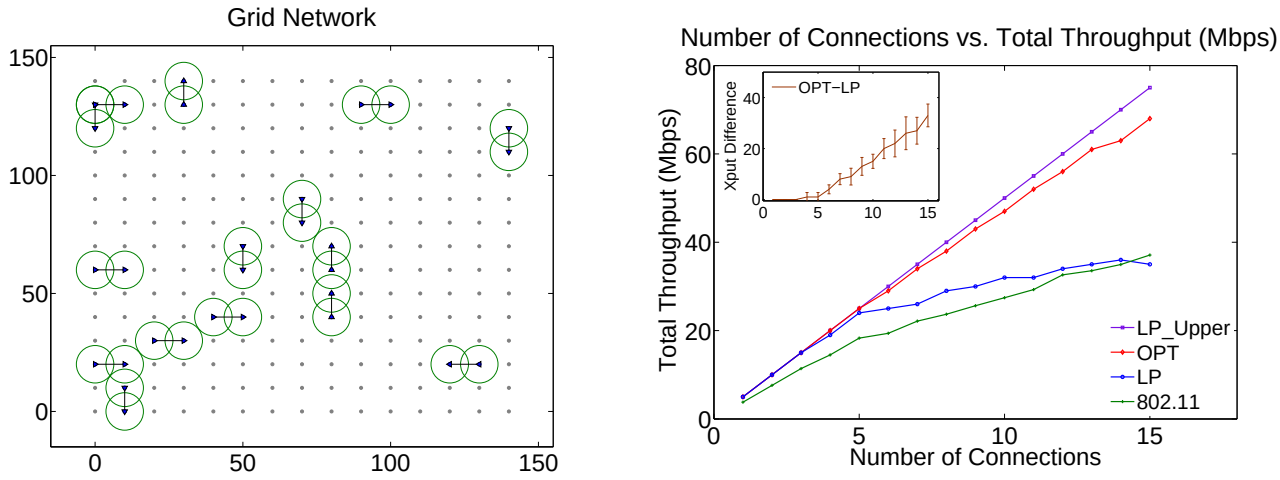


Figure 5.5: Variation in queue size w.r.t simulation time. For the 802.11 protocol, queue size is observed at a particular node.

- *Impact of congestion:* We observe for all the three network types (cf. Figure 5.4(b), 5.6(b),5.7 (b)) that as the number of connections in the system increase the LP throughput drifts significantly from the optimal solution. These results suggest that our proposed congestion measure (cf. Chapter 2 Section 2.5.4) can yield pessimistic estimates for high traffic regimes. Our techniques based on the congestion measure may prevent certain links from transmitting simultaneously which could potentially be scheduled at the same time.

- *Impact of length diversity:* For the case of non-uniform edge lengths (which is the case for random and realistic network), the LP scales down the overall rates by a scaling factor of $1 + \log \Delta$, where Δ denotes the maximum inter-point separation (ratio of the maximum length edge and minimum length edge). Recall that in our approximation scheme, we partition the edges into $\log \Delta$ bins based on their edge lengths, such that each bin $H^i = \{e \in L : \ell(e) \in [2^i, 2^{i+1})\}$. We then consider each bin separately and schedule all the links belonging to it (cf. Chapter 4, Section 4.6.3). For the cases of random and realistic networks, the average length diversity was 5.2 and 6.15



(a) (b)

Figure 5.6: Simulation results for a grid network consisting 15×15 nodes, with uniform grid spacing of $10m$. There are no fairness constraints. (a) Instance of the grid network for a particular seed. One-hop source-destination pairs are selected u.a.r. (b) Variation of throughput w.r.t. the number of connections as observed by the optimal solution, LP and 802.11 protocol. LP_Upper denotes the upper bound computed by running the LP with the necessary conditions. The 802.11 protocol ran without rts-cts and the queue size was fixed to 12.5MB. The total number of bins in the LP was set to one.

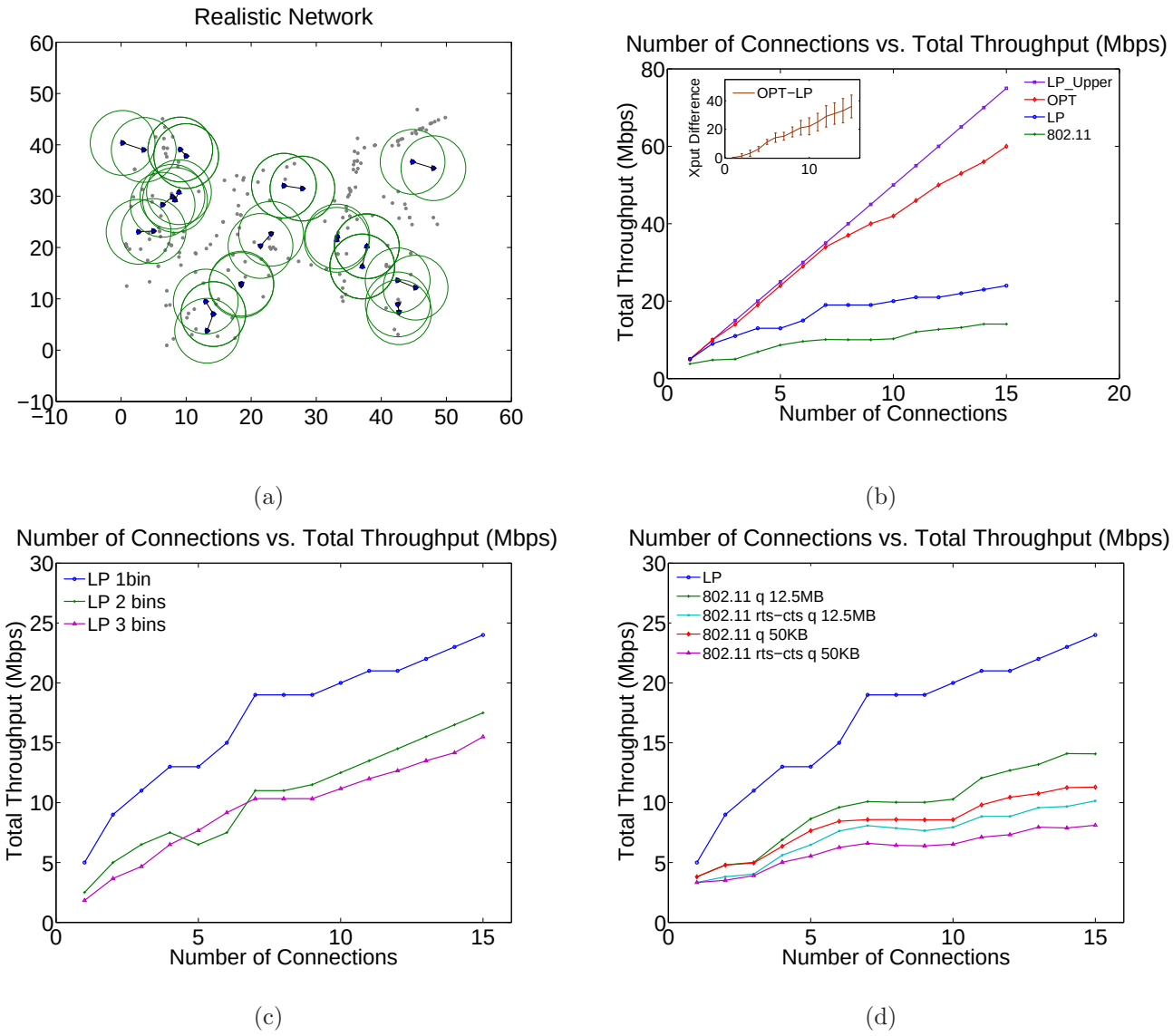


Figure 5.7: Simulation results for a realistic network consisting of 227 nodes in a $50m \times 50m$ square area. There are no fairness constraints. (a) Instance of the realistic network for a particular seed. One-hop source-destination pairs are selected u.a.r. (b) Variation of throughput w.r.t. the number of connections as observed by the optimal solution, LP and 802.11 protocol. LP_Upper denotes the upper bound computed by running the LP with the necessary conditions. The 802.11 protocol ran without rts-cts and the queue size was fixed to 12.5MB. The total number of bins in the LP was set to one. (c) Different binning strategies were considered for the LP. (d) 802.11 protocol was considered with and without rts-cts messaging, for different queue sizes, 12.5Mbytes and 50Kbytes respectively.

suggesting the use of 3 bins. We observed that (cf. Figures 5.4(c) and 5.7(c)) this binning strategy highly underestimates the LP throughput. We therefore experimented with different bin-widths. By increasing the width of each bin by a quantity p, each bin can accommodate more links making $H^i = \{e \in L : \ell(e) \in [2^i, 2^{i+p})\}$. This results in fewer bins and hence, the throughput scaling factor can be reduced (the scaling factor is equal to the total number of bins). In order to ensure that the schedule is feasible by increasing the bin width, we impose additional conditions on the value of constant a (cf. Chapter 4, Section 4.6.3). For the case of uniform power levels, the new constraints on the value of a are of the form $a \geq 2\sqrt[\alpha]{\frac{3\cdot 2^{2p+2}\beta(1+\epsilon)}{\epsilon(\alpha-2)}}$, where p denotes the increase in the bin-width, $p = 1$ implies the original case. This condition can be derived by following the proof for Lemma 4.4 in Chapter 4. It should be noted that there is a tradeoff between p and a. Increasing the bin width reduces the scaling factor but increases the lower bound on the value of a. The interference set of every edge $C(e)$ is based on the value of a. Recall that

$$\forall e \in \mathcal{E},\ C(e) = \{e' = (u', v') \in \mathcal{E} : a \cdot d(u', v') \geq d(u, u') \bigwedge d(u', v') \geq d(u, v)\}.$$

As the value of a increases, more links can be included in the set $C(e) \forall e \in \mathcal{E}$, resulting in a lower throughput. We experimented with different bin-widths p (and hence different number of bins) and different values of a and observed that the overall throughput is always high when there is only a single bin (which implies that the scaling factor is 1) in the system (cf. Figures 5.4(c) and 5.7(c)).

- *Impact of topology:* We observe that for the grid, random, and realistic networks, the approximation algorithm on average performs within 68%, 42% and 48% of the optimal solution, respectively. The main reason that the approximate algorithm performs better on the grid as opposed to other networks is length-diversity. For the uniform spacing grid all the links have the same length (the transmission range and the grid-spacing is set to 10m) which is not the case for other networks. Even though we neglect the effect of binning and consider a single bin for all three network types, the value of

a is higher for the random and realistic network than the grid network.

- *Approximation Ratio:* We compute the approximation ratio for all three network types. In Table 5.2, we compare this approximation ratio with the theoretically estimated approximation ratio (ratio of the upper bound and lower bound) for the grid, random and realistic networks. We observe that the upper bound as determined by the LP monotonically increases with the number of connections and has a value of $(k \cdot cap)$, where k denotes the number of connections and $cap = 5Mbps$ is the bandwidth of the system (cf. Figure 5.4(b), 5.6(b),5.7(b)). The lower bound on the other hand is 1 as we use a single bin in the simulations. We observe that the approximation ratio is much lower than the estimated theoretical approximation ratio. This shows that in practice our approximation techniques provide a better approximation ratio than predicted and the bounds derived by our methods are indeed worst-case bounds. This result indicates that the derived upper and lower bounds are weak and additional research is required to improve these bounds.

- *Comparison with 802.11* For all three network types, we observe that the results of our approximation techniques are comparable with the 802.11 protocol (cf. Figure 5.4(b), 5.6(b),5.7(b)). 802.11 is a distributed random-access protocol and its performance is very close to our centralized technique. However, it should be noted that in order to ensure a fair comparison, the 802.11 simulations were conducted with a fairly high queue size of 12.5MB ($\approx$ 8000 packets). For 802.11 simulation, the rate at which packets arrive at every transmitter was set to the bandwidth (5Mbps in this case); this ensured that the MAC layer always has a packet to send. In order to prevent packet loss due to saturated queues, the queue sizes for all transmitters were set to a high value. We observed that the 802.11 throughput decreased significantly when the queue sizes were set to the most commonly used value of 50KB [32] (cf. Figure 5.4(d), 5.7(d)). In Figure 5.5 we demonstrate the increase in the queue size at a particular node with simulation time.

Conclusion: We conclude that:

1. The bounds derived by our techniques are indeed worst case approximation bounds and in practice our methods perform better than predicted. Additional research is however required to improve the upper and lower bounds.

2. The performance of the approximation algorithm is influenced by the length-diversity. For special topologies such as the grid network where all edges have equal lengths, the performance of the approximation technique was higher than the generic case.

3. For the case of random networks and the generic case, higher performance gains can be achieved by engineering the system. Partitioning the edges into different bins provided less throughput gains than using a single bin.

4. The congestion measure proposed in this dissertation can lead to overly pessimistic estimates for high traffic regimes. Additional research is required in developing an efficient congestion measure.

5.4 Impact of Fairness Constraints

Goal: The goal of this set of experiments is to study the variation of the overall throughput as a function of end-to-end fairness.

We add fairness constraints to the LP as discussed in Chapter 4, Section 4.6 by introducing a quantity called fairness index λ. Recall that $\lambda = 0$ implies that there is no fairness in the system, and complete starvation of flows would be allowed. $\lambda = 1$ on the other hand implies that all the flows have identical throughput. In this set of experiments, we do not consider the impact of routing; for a given set of randomly chosen source destination pair, we compute the shortest path, using the Dijkstra's algorithm. The shortest path routes chosen for a particular seed are shown in Figure 5.2.

Hypothesis: We expect the overall throughput to decrease as the fairness index approaches one.

Results and Explanation: Figures 5.4(a),(b),(c) plot the results of our experiments for different network types. It can be seen that the overall throughput decreases monotonically as a function of the fairness index λ. We observe a sharp decline in throughput when fairness constraints are added. This suggests that maximum network throughput is achieved at the cost of assigning a rate of zero to certain end-to-end flows. The "fair" ($\lambda = 1$) throughput is less than half the maximum total throughput ($\lambda = 0$).

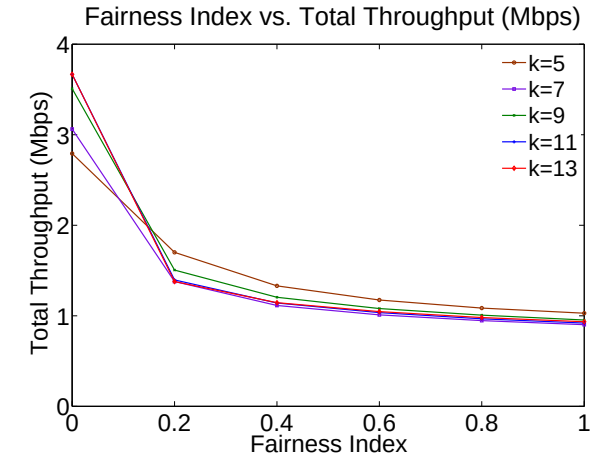
Conclusion: We conclude that the fairness constraints heavily influence the overall throughput derived by the LP and starvation of flows is possible in the absence of the fairness constraints.

5.5 Impact of Cross-layer interaction

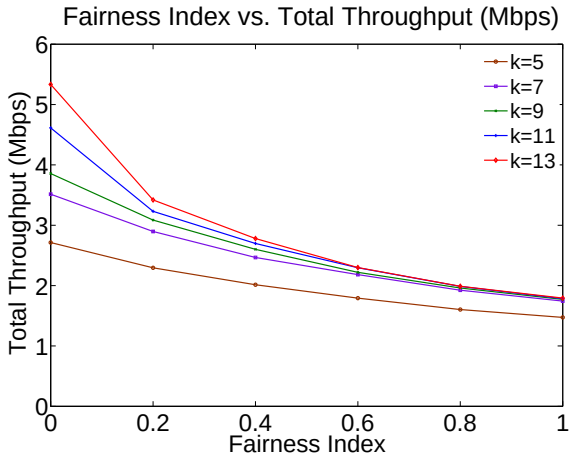
In all the experiments discussed so far, we do not consider the effect of routing. We now discuss the impact of congestion-aware routing on the system throughput. Congestion-aware routing selects routes based on the interaction with the MAC layer in order to reduce the overall MAC-level congestion. In this set of experiments, we propose a routing metric based on our congestion measure (cf. Chapter 2, Section 2.5.4).

Goal: The goal of this experiment is to study the impact of congestion-aware routing on the overall throughput

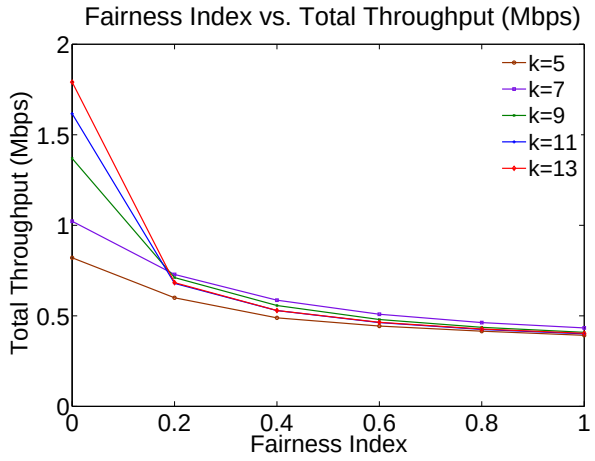
The LP formulation discussed in Chapter 4, Section 4.6 constructs multi-path routes that can reduce MAC-level congestion. Implementing these routes in a realistic setting however requires additional overhead in terms of route maintenance, size of the routing table, etc. Motivated by the work of Kumar *et al.* [30], we propose a congestion-aware path selection



(a) Geometric Random Network



(b) Grid Network



(c) Realistic Network

Figure 5.8: Variation of throughput w.r.t. the fairness index for different connections k. (a) Geometric Random Network consisting of 225 nodes in a 50×50 square area. (b) Grid Network consisting of 225 nodes, with uniform grid spacing of 10 units. (c) Realistic Network consisting of 227 nodes in a 50×50 square area.

strategy that can convert the LP based multi-path routing solution into a single path solution. We associate a weight $w(e)$ with each link e in the network that is related to the congestion $C(e)$ of the link, such that $w(e) = \gamma \cdot C(e)$, where γ is a parameter. The total weight of a path is the sum of the weights of the edges that lie on the path. Initially, all the edges have $w(e)$ set to 1. For a given set of source-destination pairs, we randomly select a connection pair and select a minimum weight route. We then update the weights of all the edges in the selected path. This procedure is repeated for all source-destination pairs. Procedure PathSelection describes that path-selection technique in detail. By selecting minimum weight paths, we intend to avoid interference due to transmissions on links that are close to each other. We compare the impact of our path selection strategy with a hop-count based metric for selecting routes. This metric selects shortest paths for a given source-destination pair based on the number of hops. We study the variation of the overall throughput as a function of the number of connections. We repeat this experiment for different network types. We do not add fairness constraints.

Hypothesis: We expect our congestion-aware metric to perform better than the hop-count based method.

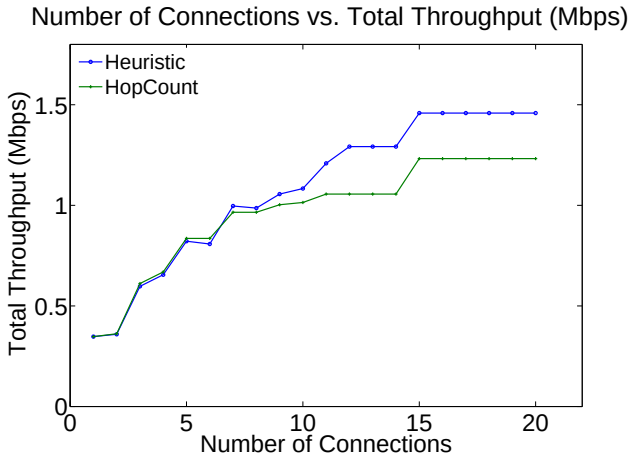
Results and Explanation: Figures 5.9(a),(b),(c) plot the results of our experiments for different network types. We observe that the congestion based path selection heuristic overall performs better than the hop-count based for all three network types. We observe a particularly large difference in the throughput for the random network. It can be seen that as the number of connections increase, the throughput difference between the congestion based strategy and the hop-count based shortest path algorithm increases. This is because, as there are more connections, the congestion in the system increases, and our metric tries to avoid high congestion paths as much as possible. We also observe a large difference in the throughput for the grid and the realistic network. This is likely due to the fact that the many links in the system could be high congestion links and may have a high weight $w(e)$ associated with them. The total weight of the shortest path may not be significantly different than minimum weight path.

Algorithm 3: PathSelection

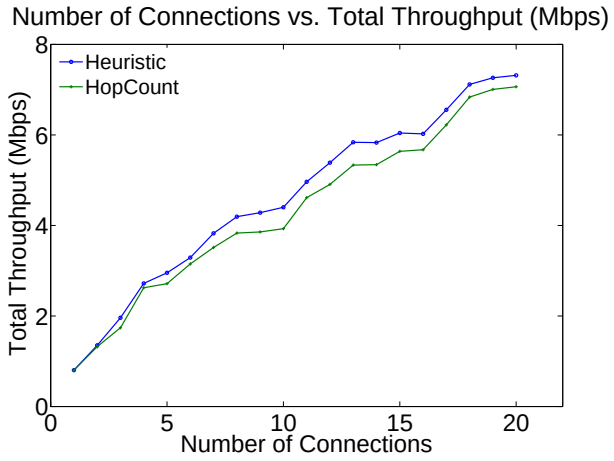
Input : (i) Graph(V,E), (ii) Set $\mathcal{K}$ of k connections , (iii) γ

Output : $\mathcal{P} = \{P_1, \ldots, P_k\}$ set of paths for every connection

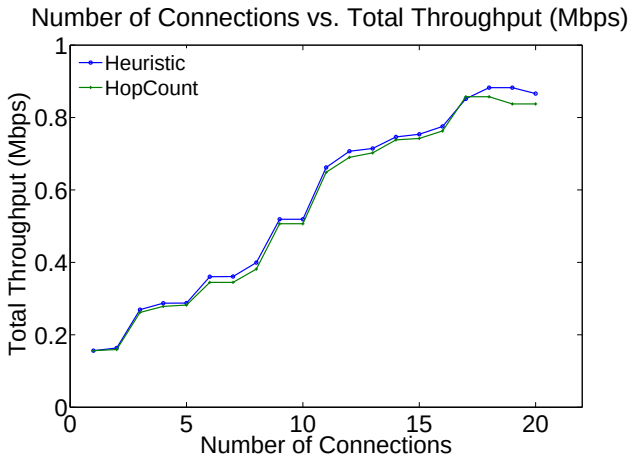
```
1  D = K
2  for e in E do
3  |   w(e) = 1
   |   //Initialize edge weights
4  end
5  for i = 1 to k do
6  |   u.a.r select source-destination pair j from D
7  |   D = D \ j
8  |   P_j = ShortestPath(j)
   |   //Compute Shortest Path based on the edge weights
   |   //E(P_j) denotes the edges belonging to path P_j
9  |   for e ∈ E do
10 |   |   for e' ∈ E(P_j) and e ≠ e' do
11 |   |   |   if e ∈ C(e') or e' ∈ C(e) then
12 |   |   |   |   w(e) = w(e) + 1
13 |   |   |   end
14 |   |   end
15 |   |   w(e) = γ · w(e)
16 |   end
17 end
```



(a) Geometric Random Network



(b) Grid Network



(c) Realistic Network

Figure 5.9: Variation of throughput w.r.t. the number of connections for congestion-aware path selection heuristic and hop-count based shortest path algorithm. There are no fairness constraints. (a) Geometric Random Network consisting of 225 nodes in a 50×50 square area. (b) Grid Network consisting of 225 nodes, with uniform grid spacing of 10 units. (c) Realistic Network consisting of 227 nodes in a 50×50 square area.

Conclusion: We conclude that our congestion-aware metric can be an effective alternative to simple hop-count based shortest path routing schemes.

5.6 Summary

We conducted extensive simulations to validate our theoretical model for the throughput maximization problem with SINR constraints. We evaluated the performance of our approximation techniques by comparing them with the optimal solution. We observed that in practice our methods can provide better bounds than predicted. The experimental evaluation allowed us to understand the limitations of our model and assisted in identifying parameters for improving the performance of our model. We studied the impact of fairness constraints on the overall throughput. Finally, we studied the impact of congestion-aware routing and observed that performance improvement is possible by enabling interactions between the MAC and the routing layers.

Chapter 6

Power Efficient Throughput Maximization in Multi-hop Wireless Networks

6.1 Problem Description

Power usage is usually a severe constraint in multi-hop wireless networks, because of limited infrastructure support. This makes the issue of maximizing the total throughput rate, while conserving the total power consumed an important problem; see [2] for a comprehensive survey. We study the problem of joint optimization of the total throughput rate and the total power used in an arbitrary multi-hop wireless network, in which nodes have the capability to adaptively choose power levels. Ongoing advances in software defined radio (SDR) technology allow adaptive power control with low overhead, making the problem of choosing suitable power levels a very important and timely issue. Formally, given (i) a set of nodes V and a set $\mathcal{E} = V \times V$ of possible directed edges, (ii) a set of connections $\mathcal{K} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$, (iii) a set $\mathcal{J}$ of power level choices, and (iv) a bound B on the total power consumed, the

objective of the *Power Efficient Throughput Maximization* (PETM) problem is to maximize the total throughput rate and ensure that the total power used is at most B, by choosing: (i) routes for each connection, (ii) an interference free schedule for all the links, (iii) power level for each link at each time slot, and (iv) data rates for each connection. We study this problem with the disk-graph interference models.

For the PETM problem, individual link capacities are affected by variations in power levels. In addition, the scheduling and power control steps are coupled in a complex manner because of the interference. The set $I(e)$ (set of edges that interfere with e), depends not only on the power level used on e, but also on the power levels used on other nearby simultaneously transmitting edges. It can be seen from Figure 6.1 that by changing the power level on edge $e1 = (u, v)$ from p_1 to p_2, the interference sets $I(e_1)$ and $I(e_3)$ are altered. Most of the prior work on provable algorithms for power control has ignored its adaptive aspect. One of the papers that is most closely related to this work is by Bhatia and Kodialam [5], who study the PETM problem with the assumption that the interference graph (the graph that specifies the set of edges interfering with any edge) is fixed initially, i.e., varying the power levels only changes the link capacities, but not the interference graph. An important consequence of these assumptions is that the power level for any link in the optimum solution does not vary with time and, therefore, it suffices to focus on non-adaptive power control. This is no longer true for the PETM problem in the general setting - we show that there are instances in which the transmission power levels on edges have to be changed adaptively in order to achieve the optimal throughput rate, and there is a significant reduction in the capacity by requiring fixed power levels for edges (cf. Figure 6.1). The non-adaptive power control problem has been shown to be non-convex by Bhatia and Kodialam [5]. Adaptive power control increases the search space of the PETM problem, thereby making it a complex and challenging problem to solve in polynomial time. We therefore focus on developing rigorous polynomial time algorithms with provable performance guarantees.

6.2 Overview of Results

The main results obtained for the PETM problem are summarized as follows:

- We develop a linear programming (LP) formulation to approximate the PETM problem, which incorporates constraints from the physical, MAC, and routing layers. Our formulation guarantees that the total throughput achieved is at least r_{opt}/μ, where r_{opt} is the optimum throughput rate and μ is a constant which depends on the interference model used. This LP formulation can have exponential size, and our algorithm involves another relaxation, which can be solved in polynomial time, and guarantees a throughput of at least $r_{opt}/(1+\epsilon)\mu$, where $\epsilon > 0$ is any constant - the running time is polynomial in $1/\epsilon$.

- We conduct simulations to study the explicit tradeoffs between quantities such as the total throughput, total power used, fairness index, and the number of connections. We observe empirically that adaptive power control leads to a much higher total throughput as compared to the non-adaptive setting. We observe that by enforcing fairness among various connections, the throughput varies significantly. The total throughput reaches a saturation point as the total power limit is increased. This allows us to understand the capacity limitations of the system.

6.3 Overall Approach

In this section, we outline the overall strategy used to obtain a linear approximation to the non-convex PETM problem.

- *Mathematical Modeling*: The exact mathematical formulation of the PETM problem yields a *non-convex* rate region due to the non-linear relationship between power and

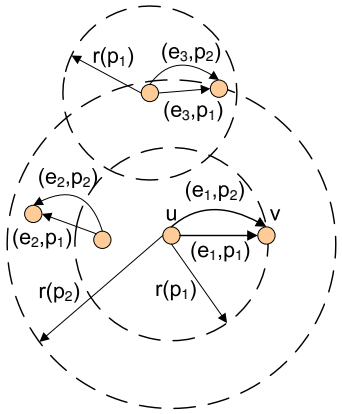


Figure 6.1: Illustrating change in the interference graph with change in transmission power levels. Each node has two possible power levels p_1 and p_2, and let c_1 and c_2 denote the link rates corresponding to these power levels. The rings of radii $r(p_1), r(p_2)$ represent transmission ranges for node u for power levels p_1, p_2. Using the Tx-model, we have, $I(e_1, p_1) = \{(e_1, p_2), (e_2, p_1), (e_2, p_2)\}$, $I(e_1, p_2) = \{(e_1, p_1), (e_2, p_1), (e_2, p_2), (e_3, p_1), (e_3, p_2)\}$, and $I(e_3, p_1) = \{(e_1, p_2)\}$. Suppose we want to maximize the flow on e_1, and require a rate of $c_1/3$ on link e_3. Then, the optimal solution is to have link rates $x(e_1, p_1) = x(e_3, p_1) = 1/3$ and $x(e_1, p_2) = 1/3$, which leads to a total rate of $c_1/3 + 2c_2/3$ on e_1, instead of c_1 or $2c_2/3$, which is possible by fixing the power level on e_1 to be fixed at p_1 or p_2, respectively.

capacity. Therefore, we carefully re-model the PETM problem to obtain a linear programming formulation. Instead of considering capacity and power variables (as considered in [5]), we consider tuples (e, p) involving an edge and the transmission power level on it (cf. Figure 6.1). The capacity of each edge from the tuple (e, p) depends on its transmission power p. For a given set $\mathcal{J}$ of power level choices, the PETM problem simply reduces to choosing a particular power level (and hence the corresponding edge) for a given node at a given time.

- *Linear Relaxation*: We utilize linear relaxation techniques from [30] to relax the scheduling constraints and obtain the necessary and sufficient conditions to approximate the rate region.

- *Discretization*: The linear program derived could be exponential in number of variables and constraints, if the set $\mathcal{J}$ of possible power levels is a large set. In order to overcome this difficulty, we develop an approximation procedure that discretizes the power levels from the set $\mathcal{J}$ in order to obtain a polynomial sized linear program.

6.4 Notation

In this section, we briefly describe the notation used in this chapter. Let V denote the set of transceivers and $\mathcal{E} = \{(u, v) \in V \times V\}$ be the set of all possible directed edges. Let K denote the set of connections and let $\mathcal{J}$ denote a set of possible power levels a node $u \in V$ can choose to transmit at. Let $\mathcal{T} = \mathcal{T}(\mathcal{J}, \mathcal{E}) = \{(e = (u, v), p) \in \mathcal{E} \times \mathcal{J} : d(u, v) \leq range(p)\}$ denote the set of all possible feasible transmission tuples. We define $N_{out}(u) = \{(e = (u, v'), p) : e \in \mathcal{E}, p \in \mathcal{J}\}$ and $N_{in}(u) = \{(e = (v', u), p) : e \in \mathcal{E}, p \in \mathcal{J}\}$ as a set consisting of outgoing and incoming transmissions to node $u \in V$ respectively.

Interference Model: We extend the usual notion of interference among links to that involving transmission tuples. For edges $e = (u, v), e' = (u', v')$, the tuples (e, p) and (e', p') interfere with each other if either (i) node u' or v' is within *interference range* of u, or (ii) node u

or node v is within interference range of node u'. In the Tx-model, the interference range associated with (e,p) is $(1+\Delta)range(p)$ [30], where Δ is a constant parameter. For $e=(u,v)$, we define $I(e,p) = I_{pri}(e,p) \cup I_{sec}(e,p)$ to be the set of transmission tuples that interfere with (e,p); this set is partitioned into primary and secondary interference sets $I_{pri}(e,p)$ and $I_{sec}(e,p)$. The primary interference set for (e,p) consists of transmissions that share a common end-point with the edge e. $I_{pri}(e,p) = \{(e'=(u',v'),p') : e' \in V \times V,\ p' \in \mathcal{J} \wedge e' \in N(u) \cup N(v)\}$. We consider the secondary interference set for any (e,p) according to the Tx-model. $I_{sec}(e,p) = \{(e'=(u',v'),p') : e' \in V \times V,\ p' \in \mathcal{J} \wedge (e',p') \notin I_{pri}(e,p) \wedge d(u,u') \leq (1+\Delta)(range(p)+range(p'))\}$. Let $I_{sec\geq}(e,p) = \{(e'=(u',v'),p') \in I_{sec}(e,p) : p' \geq p/\theta\}$ where $\theta \geq 1$. We set $\theta = 2$. Let $I_{\geq}(e,p) = I_{pri}(e,p) \cup I_{sec\geq}(e,p)$ (cf. Fig. 6.1). Note that the interference relations are symmetric, therefore if $(e',p') \in I(e,p)$ then $(e,p) \in I(e',p')$. Similar secondary interference sets can be defined for the Protocol model [30].

Link rates and feasible schedules. We assume that time is divided into slots of uniform length τ, the system operates in a synchronous mode and the transmissions occur on an error-free channel. Let r_i denote the rate on connection i, and $r = \sum_{i=1}^{k} r_i$ denote the total rate (throughput) achieved over all the k connections. We define flow and utilization vectors over transmission tuples instead of links. Let $f_j(e,p)$ denote the mean transmission rate on the transmission tuple (e,p) for connection j, and let $f(e,p) = \sum_j f_j(e,p)$ - this is the mean rate at which link e transmits using power level p. Let $x(e,p) = f(e,p)/cap(e,p)$ denote the mean utilization fraction for this transmission tuple, and let $\bar{x}$ denote the utilization vector. The (long term) fairness index $\gamma \in [0,1]$ is defined to be the ratio between the minimum and maximum rates, i.e. $\gamma = \dfrac{\min_i r_i}{\max_i r_i}$. An end-to-end schedule $\mathcal{S}$ describes the specific times at which transmissions occur. Let $X(e,p,t)$ be an indicator variable which is 1 if the transmission (e,p) is scheduled by $\mathcal{S}$ at time t. Then, the utilization fraction $x(e,p)$ corresponding to this schedule is $x(e,p) = \lim_{T\to\infty} \sum_{t\leq T} X(e,p,t)/T$. The schedule $\mathcal{S}$ is interference-free if for each time t, for each $(e,p) \in \mathcal{T}$, $X(e,p,t) = 1$ and $X(e',p',t) = 0$ for all $(e',p') \in I(e,p)$. We say that a utilization vector $\bar{x}$ is feasible if there is an interference-free schedule $\mathcal{S}$ that achieves it. The objective of the PETM problem is to find a feasible

utilization vector $\bar{x}$ so that the total throughput rate $\sum_i r_i$ corresponding to it is maximized and the total power consumed is within a certain bound B.

6.5 An approximation algorithm for the PETM problem

We utilize the linear relaxation technique discussed in Chapter 4, in order to obtain a linear programming formulation for the PETM problem. As explained previously, we derive upper and lower bounds on the rate regions.

6.5.1 Link Rate Stability: Necessary Conditions

In this subsection we derive an upper bound on the rate region, for graph-based interference models.

Lemma 6.1. *Any feasible link utilization vector $\vec{x}$ satisfies the following constraints, $\forall (e,p) \in \mathcal{T}$,*

$$\sum_{(e',p') \in I_{pri}(e,p)} x(e',p') + \sum_{(e',p') \in I_{sec\geq}(e,p)} x(e',p') \leq \mu.$$

Proof: Consider a valid schedule $\mathcal{S}$. As defined earlier, let $X(e,p,t)$ be the indicator variable for this schedule, which is 1 if the transmission tuple (e,p) is scheduled in $\mathcal{S}$ at time t. A key property of any feasible schedule is:

$$\forall (e,p) \in \mathcal{T}, \forall t, \sum_{(e',p') \in I_{sec\geq}(e,p)} X(e',p',t) \leq \mu,$$

where μ is a fixed constant that depends only on the interference model; for the Tx-model, $\mu = 5$. We skip the proof of this property, which is based on the techniques developed in [30].

Next, during any time slot, exactly one of the following events occur:
(1) Transmission (e,p) is active: In this case none of the other transmissions (excluding

transmission (e,p)) from the set $I_{pri}(e,p) \cup I_{sec\geq}(e,p)$ can be active.
(2) Transmission (e,p) is inactive: From the above property, at most μ transmission tuples from the set $I_{sec\geq}(e,p)$ can be active at time t. Combining all of the above, we get, $\forall (e,p) \in \mathcal{T}, \forall t$,

$$\sum_{(e',p') \in I_{pri}(e,p)} X(e',p',t) + \sum_{(e',p') \in I_{sec\geq}(e,p)} X(e',p',t) \leq \mu.$$

Therefore, for any time T, we have, $\forall (e,p) \in \mathcal{T}$,

$$\sum_{t \leq T} (\sum_{(e',p') \in I_{pri}(e,p)} X(e',p',t) + \sum_{(e',p') \in I_{sec\geq}(e,p)} X(e',p',t)) \leq \mu T.$$

Dividing both sides by T, and using the fact that $\lim_{T \to \infty} \sum_{t \leq T} X(e,p,t)/T = x(e,p)$, the Lemma follows.

Note that the constraint in the above lemma involves only the transmission tuples in $I_{sec\geq}(e,p)$, instead of $I(e,p)$ - this is crucial, because it is easy to construct instances in which $\sum_{(e',p') \in I_{pri}(e,p)} x(e',p') + \sum_{(e',p') \in I_{sec}(e,p)} x(e',p')$ is unbounded.

6.5.2 Link-Rate Stability: Sufficient Conditions

Utilization vectors $\bar{x}$ satisfying the constraints of Lemma 6.1 are not necessarily feasible, i.e., there may not exist a feasible schedule that could lead to the utilization vector $\bar{x}$. We develop conditions under which $\bar{x}$ can be scheduled feasibly. We describe a centralized greedy algorithm (cf. Algorithm 4) for constructing a feasible schedule.

We assume that time is divided into uniform and contiguous windows or frames of length w and w.l.o.g. $w \cdot x(e,p)$ is integral for all $(e,p) \in \mathcal{T}$. Algorithm Schedule constructs a periodic schedule $\mathcal{S}$, by repeating a schedule $\mathcal{S}_M$ for every frame M. Within each frame M, the algorithm assigns a set of $s(e,p)$ slots for every transmission (e,p), where $|s(e,p)| = w \cdot x(e,p)$, in such a way that no two interfering transmissions are assigned the same time slot. For example, if $(e',p') \in I_{\geq}(e'',p'')$, then $s(e',p') \cap s(e'',p'') = \phi$. The schedule $\mathcal{S}_M$ is constructed

by combining schedules from all the frames. For the algorithm to be stable, we need to find conditions under which step 7 of the algorithm would be successful. The following lemma gives sufficient conditions for feasible utilization vectors and shows that the algorithm is indeed successful.

Lemma 6.2. *Any utilization vector $\bar{x}$ satisfying the following conditions can be scheduled feasibly,* $\forall (e,p) \in \mathcal{T}$,

$$\sum_{(e',p') \in I_{pri}(e,p)} x(e',p') + \sum_{(e',p') \in I_{sec\geq}(e,p)} x(e',p') \leq 1.$$

Proof: (Sketch) Suppose step 7 of Algorithm Schedule fails for some transmission (e,p), then it must be the case that

$$\sum_{(e',p') \in I_{pri}(e,p)} x(e',p') \cdot w + \sum_{(e',p') \in I_{sec\geq}(e,p)} x(e',p') \cdot w > w.$$

Dividing both sides by w, we get $\sum_{(e',p') \in I_{pri}(e,p)} x(e',p') + \sum_{(e',p') \in I_{sec\geq}(e,p)} x(e',p') > 1$, which contradicts the condition on $\bar{x}$ and the Lemma follows.

Algorithm 4: Schedule

Input : (i) $\vec{x}$, (ii) w, (iii) M

Output: Schedule for a frame

1 **for** $(e,p) \in \mathcal{T}$ **do**

2 $\quad s(e,p) = \phi$

3 **end**

4 Order tuples in $\mathcal{T}$ in non-increasing inductive order. **for** *each* $(e,p) \in \mathcal{T}$ *in this order* **do**

5 $\quad s'(e,p) = \bigcup_{(e',p') \in I_{pri}(e,p)} s(e',p') \bigcup_{(e'',p'') \in I_{sec\geq}(e,p)} s(e'',p'')$

6 $\quad$ For $s(e,p)$ choose any subset of $M \setminus s'(e,p)$ of size $x(e,p)w$

7 **end**

6.5.3 Linear Programming Formulation for PETM

We now put together the constraints from the above sections to obtain a linear program $\mathcal{P}$ for an instance $\mathsf{PETM}(V, \mathcal{C}, \mathcal{J}, B)$. Equation 6.1, defines the link utilization for each transmission. Equation 6.2 captures the wireless interference. These constraints along with the end-to-end scheduling algorithm discussed previously ensure that flows computed by the LP are feasible. Equations 6.3 through 6.6 capture flow conservation constraints and are responsible for selecting appropriate routes. Equation 6.7 ensures that the total power consumed is within a bound B. Equation 6.8 captures the fairness constraints that ensure that all the flows get a fair share of the throughput. The linear program $\mathcal{P}$ can be efficiently solved using standard solvers such as NEOS [52] to obtain the value of r and $x(e,p), \forall (e,p) \in \mathcal{T}$.

$$\max\ r = \sum_{j \in \mathcal{C}} r_j \text{ s.t.:}$$

$$\forall (e,p) \in \mathcal{T}, x(e,p) = \frac{f(e,p)}{cap(e,p)} \tag{6.1}$$

$$\forall (e,p) \in \mathcal{T}, \sum_{(e',p') \in I_{pri}(e,p)} x(e',p') + \sum_{(e',p') \in I_{sec\geq}(e,p)} x(e',p') \leq 1 \tag{6.2}$$

$$\forall (e,p) \in \mathcal{T}, f(e,p) = \sum_{j \in \mathcal{C}} f_j(e,p) \tag{6.3}$$

$$\forall j \in \mathcal{C}, \sum_{(e,p) \in N_{out}(s_j)} f_j(e,p) = r_j \tag{6.4}$$

$$\forall j \in \mathcal{C}, \forall u \neq s_j, t_j \sum_{(e,p) \in N_{out}(u)} f_j(e,p) = \sum_{(e,p) \in N_{in}(u)} f_j(e,p) \tag{6.5}$$

$$\forall j \in \mathcal{C}, \sum_{(e,p) \in N_{in}(s_j)} f_j(e,p) = 0 \tag{6.6}$$

$$\sum_{(e,p) \in \mathcal{T}} x(e,p) \cdot p \leq B \tag{6.7}$$

$$\forall i \in \mathcal{C}, \forall j \in \mathcal{C} \backslash \{i\}, r_i \geq \gamma \cdot r_j \tag{6.8}$$

Theorem 6.1. *Let r_{opt} denote the maximum total throughput rate achievable for a given*

PETM *instance. The optimum solution* $\bar{x}$ *to linear program* $\mathcal{P}$ *is feasible and leads to a total throughput rate of at least* r_{opt}/μ*, where* μ *is a constant, depending on the interference model.*

Proof: The fact that $\bar{x}$ can be scheduled feasibly follows from Lemma 6.2. Let $\bar{x}_{opt}$ denote an optimum rate vector, which achieves a total throughput of r_{opt}. By Lemma 6.1, $\bar{x}_{opt}$ satisfies all the necessary conditions. Since all the constraints, including the power constraint, are linear, the vector $\frac{1}{\mu}\bar{x}_{opt}$ satisfies the program $\mathcal{P}$. Since $\bar{x}$ is the optimum solution to $\mathcal{P}$, it follows that the throughput achieved by $\bar{x}$ is at least r_{opt}/μ.

6.5.4 Putting everything together: a polynomial time solution

The number of variables and constraints in $\mathcal{P}$ is polynomial in the number of nodes and size of set $\mathcal{J}$, which could be unbounded, making it intractable to solve directly. We resolve this difficulty by discretizing the power levels from the set $\mathcal{J}$. We construct the set $\mathcal{J}' = \{j_1 = j_{min}, j_2 = 2j_1, \ldots, j_{k+1} = 2^k j_1\}$, where k is such that $2^{k-1}j_1 < j_{max} \leq 2^k j_1$, where j_{max} and j_{min} refer to the maximum and minimum power levels from the set $\mathcal{J}$. We assume that $j_{max}/j_{min} = \epsilon^{O(n)}$, for $\epsilon \geq 1$. We choose $\epsilon = 2$.

Theorem 6.2. *The optimum solution to* PETM$(V, \mathcal{C}, \mathcal{J}', B)$ *described above gives a epsilon* $\cdot \mu$ *factor approximation to the* PETM$(V, \mathcal{C}, \mathcal{J}, B)$ *problem, where* $\epsilon = 2$.

Proof: Let r_{opt} denote the optimum solution to the PETM problem and let $\vec{r}_{opt}$ be the corresponding rate vector. Let $\mathcal{T} = \mathcal{T}(\mathcal{J})$ and $\mathcal{T}' = \mathcal{T}(\mathcal{J}')$. We first transform the rate vector $\vec{r}_{opt}$ to another rate vector $\vec{z}$ in the following manner: (i) for each $(e,p) \in \mathcal{T}'$: define $g(e,p) = \sum_{p' \in [p,2p)} f(e,p')/(2\mu)$, (ii) for each $(e,p) \in \mathcal{T}'$: $z(e,p) = g(e,p)/cap(e,p)$.
We argue that $\bar{z}$ is a feasible solution to $\mathcal{P}(V, \mathcal{C}, \mathcal{J}', B)$, and achieves a rate of at least $\frac{\bar{r}_{opt}}{2\mu}$. This implies that this program has a non-empty solution space, and the theorem follows.

First, observe that since $cap(e,p) = W \log_2(1 + \frac{p}{N_0 W})$ (cf. Chapter 2), we have, $cap(e,p) \geq cap(e,p')/2$, if $p \leq p' \leq 2p$. Therefore, for any $(e,p) \in \mathcal{T}'$, we have,

$$\begin{aligned}\sum_{p' \in [p,2p)} x(e,p')p' &\geq \sum_{p' \in [p,2p)} \frac{pf(e,p')}{cap(e,p')}, \\ &\geq \mu z(e,p)p, \\ &\geq z(e,p)p.\end{aligned}$$

Since the expression for total power used by $\vec{z}$ is $\sum_{(e,p)\in\mathcal{T}'} z(e,p)p$, it follows from the above sequence of inequalities that $\sum_{(e,p)\in\mathcal{T}'} z(e,p)p \leq B$. It remains to show that $\vec{z}$ satisfies all the sufficient conditions for $\mathsf{PETM}(V, \mathcal{C}, \mathcal{J}', B)$. For any fixed $(e,p) \in \mathcal{T}'$, we observe that

$$\mu \geq \sum_{p' \in [p,2p)} x(e,p') \geq \sum_{p' \in [p,2p)} \frac{f(e,p')}{2cap(e,p)} \geq \mu z(e,p). \tag{6.9}$$

The first equality above follows from the necessary condition corresponding to (e,p), since $\bar{x}$ satisfies the necessary conditions - this implies that the variables $z(e,p)$ are valid utilization variables.

Next, let $\hat{E}(e,p) = \{e' : (e',p') \in I_{\geq}^{(\mathcal{J})}(e,p) \text{ for some } p'\}$ Therefore, we have

$$\begin{aligned}\sum_{(e',p')\in I_{\geq}^{(\mathcal{J})}(e,p)} x(e',p') &= \sum_{e'\in\hat{E}(e,p)} \sum_{p'\in\mathcal{J}'} \sum_{p''\in[p',2p')} x(e',p'') \\ &= \sum_{(e',p')\in I_{\geq}^{(\mathcal{J}')}(e,p)} \mu z(e',p'),\end{aligned}$$

where the first inequality follows from (6.9). Therefore, we have
$\forall (e,p) \in \mathcal{T}'(\mathcal{J}') : \sum_{(e',p')\in I_{\geq}^{(\mathcal{J}')}(e,p)} z(e',p') \leq 1$.

Therefore it can be seen the $\vec{z}$ satisfies all the sufficient conditions for $\mathsf{PETM}(V, \mathcal{C}, \mathcal{J}', B)$. Further since
$z(e,p) = \sum_{p'\in[p,2p)} f(e,p')/(2\mu)cap(e,p)$, the Theorem follows.

We summarize our approximation algorithm in Algorithm 5

Algorithm 5: ALG-PETM

Input : (i) V, (ii) $\mathcal{C}$, (iii) $\mathcal{J}$, (iv) B

Output: (i) Total throughput r, (ii) $x(e,p), \forall (e,p) \in \mathcal{T}$ (iii) Schedule $\mathcal{S}$

1 Construct set $\mathcal{J}'$ from set $\mathcal{J}$

2 Solve program $\mathcal{P}$ with set $\mathcal{J}'$ using standard LP solvers, to obtain r, and $x(e,p)$ $\forall (e,p) \in \mathcal{T}$

3 Use procedure Schedule to obtain a feasible schedule according to the link utilization

6.6 Simulations

We study the tradeoffs between various parameters for specific instances, using our algorithm for the PETM problem. Our main observations are: (i) the total throughput rate with adaptive power control is significantly higher than non-adaptive power control; (ii) the total throughput increases with the total power bound initially, but reaches a saturation point if fairness constraints are present; (iii) the system is sensitive to fairness constraints, and there is a sharp decline in the total throughput on increasing the fairness index. Also, in the presence of fairness constraints, increase in the number of connections can lead to a reduction in the total throughput beyond some point.

Experimental Setup: All the experiments were conducted on a random distribution of nodes. Except for Experiment #1, we considered a random distribution of 80 nodes on a $2000m \times 2000m$ area. Source destination pairs were chosen uniformly at random. We used $N_0 = 4 \times 10^{-10} mW/Hz, W = 100MHz, c1 = 50000, \alpha = 3$. The set $\mathcal{J}$ consisted of power-levels from $40mW - 140mW$ in increments of $20mW$. We used NEOS [52] solvers for solving all the linear and non-linear programs. We now present the objective and analysis of each experiment.

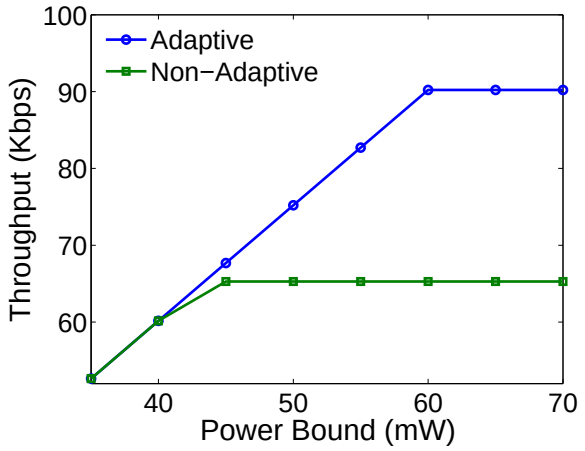
Experiment #1: Adaptive power control vs. Non-adaptive power control.
The goal of this experiment was to compare the effect of adaptive and non-adaptive power control on the total throughput rate. For the adaptive case, we solve our algorithm from

Section 6.5. We formulate the non-adaptive case as a mixed integer program, with integer variables $X(u,p)$, such that $X(u,p) = 1$ if node u chooses power level p, and solve it exactly. Since this program is difficult to solve exactly in polynomial time, we performed this experiment on a random distribution of 15 nodes on a $500m \times 500m$ area, with 2 source-destination pairs (flows) and 2 choices of power levels (40mW, 60mW) for every node. We varied the total power bound B and observed the corresponding throughput for both the adaptive and non-adaptive case. The fairness index γ was set to 0.

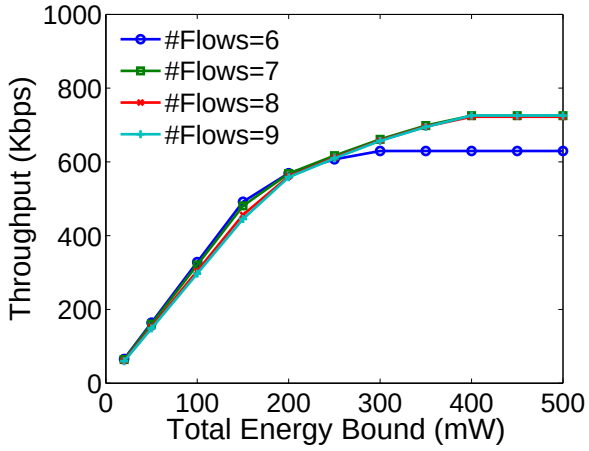
Results and Explanation: Figure 6.2(a) represents the results of our experiments. It can be seen that adaptive power control is more responsive to the changes in B than non-adaptive power control. Further the throughput achieved by adaptive power control is much higher than that achieved by non-adaptive power control, which reaches its saturation point quickly. In the case of adaptive power levels, we observe that some links choose different power levels at different times. We expect the difference in throughput to be much higher for larger networks with more flows.

Experiment #2: Throughput vs. Total power bound .
In this experiment, we varied the total power bound and observed the variation in the throughput for $\gamma = 0.01$. We expected the throughput to increase with B.
Results and Explanation: Figure 6.2(b) shows the results of our experiments for different values of flows. As expected it can be seen that the throughput increases with the power bound. The throughput however does not increase after a certain value of B and achieves saturation. At this point the system operates at its maximum capacity. This plot is useful to understand the capacity limits of the system and the tradeoff between the throughput achieved and the total power consumed.

Experiment #3:Throughput vs. Number of flows.
In this experiment, we varied the number of flows and observed the variation in the throughput for two different values of fairness index $\gamma = 0$ and $\gamma = 0.01$. Intuitively we expected the throughput to increase with the number of flows.



(a) $\gamma = 0$



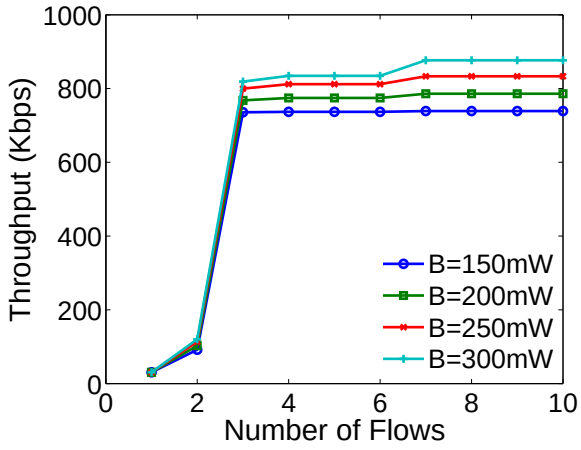
(b) $\gamma = 0.01$

Figure 6.2: Throughput(Kbps) vs. total power bound (mW). (a) Experiment #1: Adaptive vs. Non-adaptive power control. (b) Experiment #2: Throughput variation with B for adaptive power control.

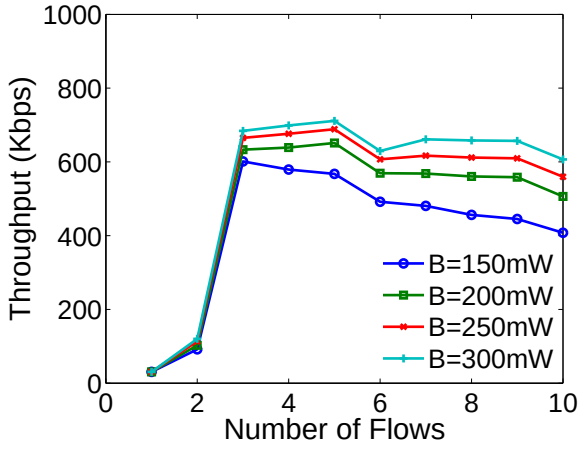
Results and Explanation: Figures 6.3(a) and 6.3(b) plot the results of our experiments for different values of total energy bound B. It can be see that for $\gamma = 0$, after a steep rise, the throughput exhibits a step like behavior and reaches saturation. The variation of throughput with the number of flows is higher for $\gamma = 0.01$. After a steep rise, the throughput starts decreasing with the number of flows. Further it can be seen that the overall throughput is higher when $\gamma = 0$ than $\gamma = 0.01$. Recall that γ is the ratio between the minimum rate and maximum rate across the flows. $\gamma = 0$ implies that there is no fairness in the system and complete starvation of flows could be allowed. $\gamma = 1$ on the other hand implies that all the flows would have identical throughput. When there is no fairness in the system, some flows are completely starved in order to achieve maximum possible aggregate throughput. The throughput does not vary even if new flows are added to the system. This is because, these new flows are assigned a rate of zero in order to maximize the overall throughput. When fairness constraints are enforced, the overall throughput can decrease in order to ensure that all the flows get their fair share of throughput.

Experiment #4:Throughput vs. Fairness index.
In this experiment, we varied the fairness index γ from 0 to 1 and observed the variation in the throughput. We fix B to $250mW$ and compute the throughput for different number of flows.

Results and Explanation: Figure 6.4 shows the results of our experiments for different number of flows. It can be seen that the throughput decreases as γ increases. There is a sharp fall in the throughput as γ varies from 0 to 0.01, and at low γ values, some flows are completely starved.

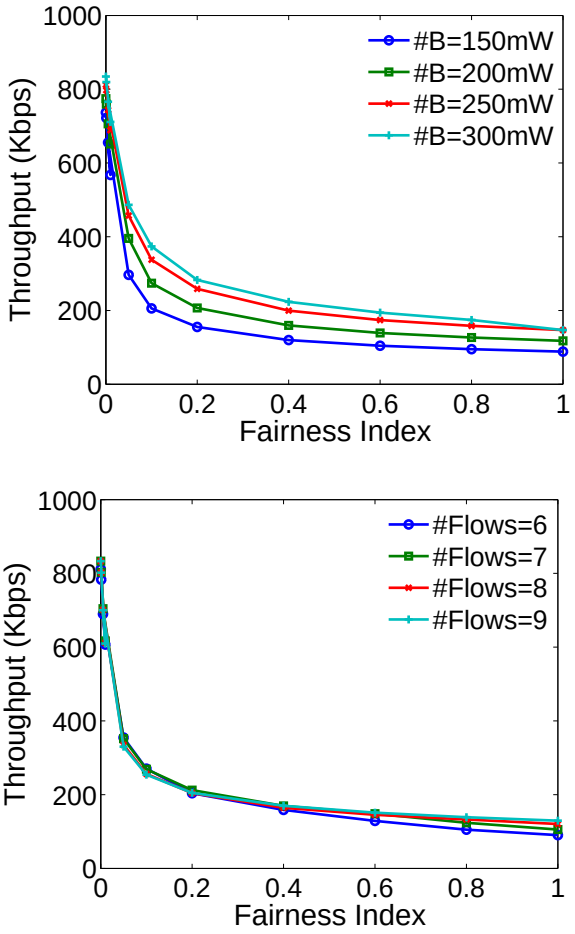


(a) $\gamma = 0$



(b) $\gamma = 0.01$

Figure 6.3: Experiment #3: Throughput (Kbps) vs. number of flows with and without fairness constraints.



(a) $B = 250mW$

Figure 6.4: Experiment #4: Throughput(Kbps) vs. fairness index

6.7 Most Relevant Prior Work

Our work with the PETM problem is most related to the work by Bhatia *et al.* [5]. The authors study the joint routing, scheduling and power control problem with the objective of minimizing total power consumed. They formulate the overall problem as a non-convex optimization problem with non-linear constraints and develop a polynomial time 3-approximation algorithm for solving this problem. There are crucial differences between our work and [5]: (i) The interference graph used in [5] is assumed to be fixed, whereas we allow it to change with the power levels, (ii) We consider general interference models that take into account both primary and secondary interference, whereas [5] only allow primary interference, (iii) our algorithm uses a linear program, which can be solved much more efficiently than the quadratic program used by [5].

6.8 Summary

We described a general algorithmic technique for solving the power efficient throughput maximization problem by jointly considering routing, scheduling and power control. Our technique guarantees that the total throughput achieved is within a constant factor away from the optimal solution. We conducted simulations to study the explicit tradeoffs between quantities such as the total throughput, total power used, fairness index, and the number of connections.

Chapter 7

Cross-layer Latency Minimization with SINR constraints

7.1 Problem Description

We consider the problem of minimizing latency (length of the end-to-end schedule) in wireless multi-hop networks. Formally, given (i) a set of nodes V and a set $\mathcal{E} = V \times V$ of possible directed edges, (ii) antenna gain β, path-loss exponent α, (iii) a set of source-destination pairs $K = \{(s_1, t_1), \ldots, (s_k, t_k)\}$, (iv) a set of packets $\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ for all k connections, and (v) a power range $[j_{min}, j_{max}]$, the objective of the *Cross-layer latency minimization* (CLM) problem is to minimize the overall latency by: (i) assigning power levels to individual transceivers, (ii) choosing routes for the packets, and (iii) constructing an end-to-end schedule for the packets, such that all the transmissions that happen simultaneously satisfy the SINR interference constraints.

The CLM problem is NP-hard to solve exactly. The proof follows by restriction: the minimum length link scheduling problem for the SINR models, in which all the connections are on adjacent links is NP-hard [19]. Since this is a special case of the CLM problem, it follows

that the CLM problem with SINR constraints is also NP-hard. Therefore, we focus on developing rigorous polynomial time approximation algorithms with provable performance guarantees.

7.2 Overview of results

The main results obtained for the CLM problem are summarized as follows:

- We describe a general algorithmic approach for designing a polynomial-time randomized approximation algorithm (MinDelay) for the CLM problem, with rigorous provable bounds on the overall latency. For a given set of n nodes in a Euclidean plane, with maximum inter-point separation of Δ (cf. Table 2.1), a given range $[j_{min}, j_{max}]$, and a small positive slack $\epsilon > 0$, our algorithm provides an end-to-end schedule, whose length is at most a $O(\log^2 n \log^3 \Delta)$ factor away from the optimum solution OPT, provided that OPT uses power from the range of $[j_{min}, (1-\epsilon)j_{max}]$. This is a worst-case approximation guarantee and holds for every instance of the problem.

- As corollaries of the CLM problem, we obtain polynomial-time randomized approximation algorithms for the following additional problems: (a) joint routing and end-to-end scheduling to minimize latency, when the power levels are pre-specified for all nodes; (b) end-to-end scheduling to minimize latency, when both the routes and power levels are pre-specified; and (c) the CLM problem with an additional constraint that the total power consumed is at most some given B: our algorithm can be modified so that the latency is optimized, with the total power used being at most $(B \log n)$, where n is the number of transceivers. It is important to note that in general, being able to solve the CLM problem efficiently *does not* immediately mean that one can solve these variants efficiently.

We view the CLM problem as a **bi-criteria** optimization problem - the two criteria being minimizing overall latency and ensuring that per node power assigned is within a given range $[j_{min}, j_{max}]$. We require a slightly higher power range than that used by the optimal solution (optimal solution assigns power levels from the range $[j_{min}, (1-\epsilon)j_{max}]$, where ϵ is a small positive slack) in order to ensure that the schedule constructed by algorithm MinDelay is feasible and the SINR constraints are satisfied at every time step. This is explained formally in Section 7.5.3 and Theorem 7.2.

7.3 Overall Approach

We develop an algorithm MinDelay (Algorithm 6) for solving the CLM problem. This algorithm can be broken down into the following main components:

- *Path Selection*: For a given wireless network, with known connections, we find a single path for every source-destination pair from a set of possible paths using the linear-programming rounding technique developed by Raghavan, and Thompson [54]. We ensure that the paths selected have a congestion value that is at most a poly-logarithmic factor away from the optimal solution.
- *Power Control*: We assign sufficient power level to every sender that lies on the selected paths such that the links on these paths are realized.
- *Scheduling*: We finally schedule packets for every connection on their respective paths by extending the random delays technique developed by Leighton, Maggs and Rao [31]. According to this technique, every packet waits for a random amount of time before being transmitted from a particular source. We conduct a detailed analysis of this technique for the wireless setting.

Although the above three components are modular, there is a significant interplay between them and they jointly minimize the end-to-end latency. The congestion measure as described

in Chapter 2, Section 2.5.4, plays an important role in enabling interactions between the routing and scheduling layers.

7.4 Notation

In this section, we briefly describe the notation used in this chapter. Let V denote the set of transceivers and $\mathcal{E} = \{(u, v) \in V \times V\}$ be the set of all possible directed edges. Let $\mathcal{K}$ denote the set of connections and let j_{min}, j_{max} denote the max and min power levels provided to the system. Let $\mathcal{Q} = \{Q_1, \ldots, Q_k\}$ denote a set of packets for k sources. Let $y(i, e)$ be a flow-indicator variable for each connection i and edge $e \in \mathcal{E}$ such that:

$$y(i,e) = \begin{cases} 1 & \text{if path for connection } i \text{ passes through } e \\ 0 & \text{otherwise.} \end{cases}$$

Let $L = \{(u, v) \in V \times V : j_{max} < (1 + \epsilon)\beta N \cdot d(u, v)^{\alpha}\}$ denote the set of links that require a power level greater than j_{max} to be realized. We treat these links as infeasible links. We recall the definitions of $N_{out}(u)$, $N_{in}(u)$, Δ, Γ_{node}, congestion C, dilation D, SINR model, and constants α, β, a from Chapter 2 and Table 2.1.

7.5 The Latency-Minimization Algorithm (MinDelay)

We now present the details of the algorithm MinDelay.

7.5.1 Path Selection

Procedure PathSelection is responsible for selecting **good** paths, i.e., paths with a "low" value of congestion and dilation ($C + D$). We formulate the path selection (routing) problem as an integer program (IP), following the methods of [54]. Note that in the path selection process we do not select edges from the set L. By forcing this constraint, we ensure that

Algorithm 6: MinDelay

Input : V, $\mathcal{E}$, K, $[j_{min}, j_{max}]$, Q, α, β

Output: (i) $\mathcal{P}$ (ii) J, (iii) A valid schedule $\mathcal{S}$ that moves packets along their paths to the destinations successfully

1 $\mathcal{P}, C = \mathsf{PathSelection}(V, K, \mathcal{E}, [j_{min}, j_{max}])$

2 $J = \mathsf{PowerControl}\ (V, K, \mathcal{P}, E(\mathcal{P}))$

3 $\mathcal{S} = \mathsf{Schedule}(V, K, \mathcal{P}, Q, J, C, E(\mathcal{P}))$

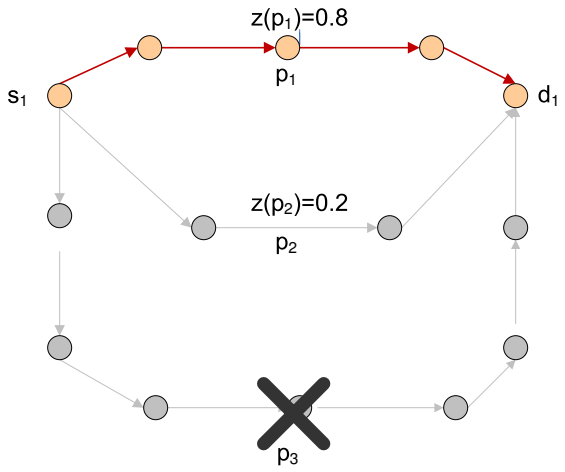


Figure 7.1: Illustration of the path selection process: For a source s_i and destination d_i, three paths p_1, p_2, p_3 are obtained as a result of the path decomposition step. The path selection process, first discards long paths, therefore path p_3 is discarded. From the remaining set of paths p_1, p_2, path p_1 is selected with probability $z(p_1)$.

Procedure PathSelection

Input : V, K, $\mathcal{E}$, and $[j_{min}, j_{max}]$

Output: $\mathcal{P}$

`//Solve the LP`

Solve the LP as described in equations 7.1..7.8 to get the solutions $(\{y_f(i,e)\}, w_f)$.

`//Path Decomposition`

Decompose the flow into path flows — let $\mathcal{M}_i$ denote the set of paths from s_i to t_i, with $h(P)$ denoting the flow on path $P \in \mathcal{M}_i$, where $h(P) = \min_{e \in P} y_f(i,e)$. Let $|P|$ denote the length (number of links) of path P. Let $\mathcal{M} = \bigcup_i \mathcal{M}_i$ denote the set of all paths from all connections.

`//Path Filtering`

foreach $P \in \mathcal{M}_i$ **do**
    $\gamma = \sum_{P' \in \mathcal{M}_i : |P'| > 2w_f} h(P')$.
    if $|P| \leq 2w_f$ **then**
        $z(P) = h(P)/(1-\gamma)$.
    else
        $z(P) = 0$
    end
end

`//Randomized Rounding`

foreach i **do**
    choose exactly one path $P \in \mathcal{M}_i$ with probability $z(P)$.
    Let $P_i = P$ be the path chosen from $\mathcal{M}_i$
end
Let $\mathcal{P} = \{P_1, \ldots, P_k\}$.

the power assigned to any link (u, v) is at least $(1+\epsilon)\beta N d(u,v)^{\alpha}$. In order to guarantee the feasibility and validity of the final end-to-end schedule, we require this condition (cf. Section 7.5.3). The following is the natural integer programming (IP) formulation.

$$\min \quad w \text{ subject to:}$$

$$\forall i \in 1, \ldots, k : \sum_{e \in N_{out}(s_i)} y(i,e) = 1 \tag{7.1}$$

$$\forall i \in 1, \ldots, k : \sum_{e \in N_{in}(s_i)} y(i,e) = 0 \tag{7.2}$$

$$\forall i \in 1, \ldots, k, \forall v \neq s_i, t_i : \sum_{e \in N_{out}(v)} y(i,e) = \sum_{e \in N_{in}(v)} y(i,e) \tag{7.3}$$

$$\forall e \in \mathcal{E} : \sum_{i=1}^{k} \sum_{e' \in C(e)} y(i,e') \leq w \tag{7.4}$$

$$\forall i \in 1, \ldots, k : \sum_{e \in \mathcal{E}} y(i,e) \leq w \tag{7.5}$$

$$\forall i \in 1, \ldots, k, \forall e \in L : y(i,e) = 0 \tag{7.6}$$

$$\forall i \in 1, \ldots, k, e \in \mathcal{E} : y(i,e) \in \{0,1\} \tag{7.7}$$

The objective of the IP formulation is to minimize the overall congestion w. Constraints (7.1) ensure that exactly one outgoing edge is selected from each source. Constraints (7.2, 7.3) capture flow conservation. Constraints (7.4) ensure that the total congestion on any link e is bounded and Constraints (7.5) guarantee that each path selected has length less than or equal to w. Constraints (7.6) ensure that the chosen paths do not include edges from set L. Let $(\{y_{int}(i,e)\}, w_{int})$ denote the integer program solution. Solving the above IP in polynomial time is difficult, and therefore we use a LP relaxation scheme, wherein we relax the constraints (7.7) to obtain constraints of the form

$$\forall i, e : y(i,e) \quad \leq \quad 1. \tag{7.8}$$

Procedure PathSelection consists of the following steps (cf. Figure 7.1).

1. Solving the LP: The LP as described above can be solved using standard LP solvers such

as NEOS [52] or CPLEX [16] to obtain the solutions $(y_f(i,e), w_f)$, $\forall e \in \mathcal{E}$. It can be seen that $w_f \leq w_{int}$.

2. Path Decomposition: With the values of $(y_f(i,e))$, we now decompose the flow into a set of paths as discussed in [54]. The basic goal of this step is to yield a set of paths $\mathcal{M}_i$ for each connection $i = 1, \ldots, k$ and assign a flow value $h(P)$ to each path $P \in \mathcal{M}_i$, such that $h(P) = \min_{e \in P} y(i,e)$. We denote $\mathcal{M} = \bigcup_i \mathcal{M}_i$ to be the set of all paths from all the connections.

3. Path Filtering: The set of paths $\mathcal{M}_i$ obtained after the path decomposition step might not satisfy the dilation constraints (Constraints 7.5). So we run a filtering step, as in [54], in order to discard long paths. After filtering out long paths, we construct a new flow variable $z(P)$ for each path $P \in \mathcal{M}_i$.

4. Randomized Rounding: After the path filtering step, we still have $|\mathcal{M}_i| > 1$, i.e., we have multiple paths for a given $s_i - t_i$ pair. We now use the Raghavan-Thompson **rounding** algorithm [54] and choose exactly one path from the set $\mathcal{M}_i$ with a probability $z(P)$ for every connection i. We thus obtain a set of paths $\mathcal{P} = \{P_1, \ldots, P_k\}$, where P_i denotes the path chosen for connection i. In the next subsection, we show that the resulting paths have congestion plus dilation of at most $O(w_{int} \log n)$.

Analysis of procedure PathSelection

We show that the paths selected by the approximation procedure described above have low values of congestion and dilation and the overall $C + D$ (congestion plus dilation) is only a $\log n$ factor away from the optimal solution with high probability.

Lemma 7.1. *In procedure* PathSelection *after the path filtering step the fractional solution z obtained satisfies the following properties:*

1. *$z(P)$ is positive only if $|P| \leq 2w_f$ for any $P \in \mathcal{M}_i$, for any i.*

2. *For each i, $\sum_{P \in \mathcal{M}_i} z(P) = 1$.*

3. *For any edge e, $\sum_{e' \in C(e)} \sum_{P \in \mathcal{M}: e' \in P} z(P) \leq 2w_f$.*

Proof: The first property is true by construction. In the path filtering step (step 8) of procedure PathSelection, $z(P)$ is set to zero for all paths having length greater than $2w_f$. Therefore for all paths having length less than or equal to $2w_f$, $z(P)$ is positive.

For the second property, from the path decomposition step for each s_i, t_i pair we have,

$$\begin{aligned} \sum_{P \in \mathcal{M}_i} h(P) &= 1 \\ \implies \sum_{P \in \mathcal{M}_i: |P| > 2w_f} h(P) &+ \sum_{P \in \mathcal{M}_i: |P| \leq 2w_f} h(P) = 1 \\ \implies \sum_{P \in \mathcal{M}_i: |P| \leq 2w_f} h(P) &= 1 - \gamma. \end{aligned}$$

By construction and the first property we have,

$$\begin{aligned} \sum_{P \in \mathcal{M}_i: |P| \leq 2w_f} z(P)(1-\gamma) &= (1-\gamma) \\ \implies \sum_{P \in \mathcal{M}_i} z(P) &= 1. \end{aligned}$$

For the third property, note that $\sum_{i=1}^{k} \sum_{e' \in C(e)} y(i, e') = \sum_{e' \in C(e)} \sum_{P \in \mathcal{M}: e' \in P} h(P)$. The path decomposition guarantees that for each s_i, t_i pair,

$$\begin{aligned} \sum_{P \in \mathcal{M}_i} |P| h(P) = \sum_{e \in \mathcal{E}} y_f(i, e) \leq w_f \\ \implies \sum_{P \in \mathcal{M}_i: |P| > 2w_f} |P| h(P) + \sum_{P \in \mathcal{M}_i: |P| \leq 2w_f} |P| h(P) \leq w_f. \end{aligned}$$

Therefore, we must have,

$$\sum_{P \in \mathcal{M}_i : |P| > 2w_f} h(P) \leq \frac{1}{2},$$

otherwise, the sum $\sum_{P \in \mathcal{M}_i} |P| h(P)$ would exceed w_f. Therefore, $\gamma \leq 1/2$. This implies that, $\forall P \in \mathcal{M}_i$ such that $|P| \leq 2w_f, h(P) \leq z(P) \leq 2h(P)$ and property 3 follows.

Theorem 7.1. *Procedure* PathSelection *constructs a set of paths $\mathcal{P}$ such that $C + D$ is $O((C_{OPT} + D_{OPT}) \log n)$ with high probability (i.e., with probability at least $1 - 1/n$), where $C_{OPT} + D_{OPT}$ denotes the smallest "congestion plus dilation" possible (provided the paths contain no edges from the set $L = \{(u, v) \in V \times V : j_{max} < (1 + \epsilon)\beta N d(u, v)^{\alpha}\}$).*

Proof: Let $y(P) \in \{0, 1\}$ be an indicator variable such that $y(P) = 1$ if path $P \in \mathcal{M}$ is selected in the rounding procedure; we have $Pr[y(P) = 1] = z(P)$ in Procedure PathSelection. Note that variables $y(P)$ are not independent. Since we only choose a single path from the set $\mathcal{M}_i$, for paths $P_1, P_2, \ldots, P_m \in \mathcal{M}_i$, w.l.o.g., if $y(P_1) = 1$ then $y(P_j) = 0, \forall j = 2, \ldots, m$. Let $\mathcal{P}(y) = \{P : y(P) = 1\}$, be the set of paths chosen in the rounding procedure. For any edge $e \in \mathcal{E}$, define $cong(e, \mathcal{P}(y)) = \sum_{e' \in C(e)} \sum_{P \in \mathcal{M} : e' \in P} y(P)$. For edge $e \in \mathcal{E}$, and path $P \in \mathcal{M}$, let $n(e, P) = |E(P) \cap C(e)|$ be the number of edges in P that lie in the set $C(e)$, where $E(P)$ denotes the edges that lie on path P. Then, we have

$$\begin{aligned} cong(e, \mathcal{P}(y)) &= \sum_{e' \in C(e)} \sum_{P \in \mathcal{M} : e' \in P} y(P) \\ &= \sum_{P \in \mathcal{M} : n(e,p) > 0} n(e, P) y(P), \end{aligned}$$

and

$$\begin{aligned} E[cong(e, \mathcal{P}(y))] &= \sum_{e' \in C(e)} \sum_{P \in \mathcal{M} : e' \in P} E[y(P)] \\ &= \sum_{e' \in C(e)} \sum_{P \in \mathcal{M} : e' \in P} z(P) \leq 2w_f, \end{aligned}$$

where the last inequality follows from Lemma 7.1. Since the coefficients $n(e, P)$ are arbitrary, we cannot directly apply the Chernoff bound to show that $cong(e, \mathcal{P}(y))$ is not much larger

than $2w_f$. However, the path filtering gives us an important property that $n(e,P) \leq 2w_f$, for all edges $e \in \mathcal{E}$, and all paths $P \in \mathcal{P}$ with $z(P) > 0$, by Lemma 7.1 - this is true, since $n(e,P) \leq |P|$. We define constants $a(e,P) = \frac{n(e,P)}{2w_f}$, and $Y(e) = \sum_{P \in \mathcal{M}:\ n(e,P)>0} a(e,P)y(P)$. Since $Y(e) = cong(e, \mathcal{P}(y))/(2w_f)$, we have $E[Y(e)] \leq 1$. Although the variables $y(P)$ are not independent, they are negatively co-related with the corresponding independent variables. Consider independent variables $x(P), \forall P \in \mathcal{M}_i$ such that $x(P) = 1$ is path P is chosen and 0 otherwise. Therefore for any $P_1, P_2, \ldots, P_m \in \mathcal{M}_i$, $Pr[x(P_1) = 1] \cdot Pr[x(P_2) = 1] \ldots \cdot Pr[x(P_m) = 1] = z(P_1) \cdot z(P_2) \ldots \cdot z(P_m)$. Let $X(e) == \sum_{P \in \mathcal{M}:\ n(e,P)>0} a(e,P)x(P)$. It can be seen that $E[Y(e)^j] \leq E[X(e)^j]$ for any j. Therefore because of negative correlation and the fact that $a(e,P) \in [0,1]$, Chernoff bound (see, e.g., page 86, [39]) can be applied to the sum $Y(e)$, and this gives

$$\begin{aligned} Pr[Y(e) \geq 3\log n] &\leq 2^{-3\log n} \\ &\leq \frac{1}{n^3}. \end{aligned}$$

This in turn implies $Pr[cong(e, \mathcal{P}(y)) \geq 3\log n \max\{2w_f, 1\}] \leq \frac{1}{n^3}$. Since the number of edges is $O(n^2)$, by union bound we get

$$Pr[\exists\ e\ s.t.\ cong(e, \mathcal{P}(y)) \geq 3\log n \max\{2w_f, 1\}] \leq \frac{1}{n}.$$

Therefore with high probability, Procedure PathSelection constructs a set of paths whose $C + D$ value is $O(w_f \log n)$, which is $O((C_{opt} + D_{opt}) \log n)$, because $w_f \leq w_{int} \leq C_{opt} + D_{opt}$. We use the value of w_f as the overall congestion measure, i.e. $\mathbf{C} = \mathbf{w_f}$. This measure will then be used by the scheduling step to estimate the random delays. We note that the PathSelection step can be derandomized using standard techniques from [48].

7.5.2 Power Control

After obtaining the set of paths $\mathcal{P}$, the power control step assigns power levels to every transmitter u such that $e = (u, v) \in E(\mathcal{P})$. Procedure PowerControl assigns power levels in

Procedure PowerControl

Input : V, K, $\mathcal{P}$, multi-set $E = E(\mathcal{P})$

Output: J

1 Choose power levels $J(u)$ for each node u as $J(u) = \max_{e=(u,v)\in E}\{(1+\epsilon)\beta N d(u,v)^{\alpha}\}$

order to physically realize all the links from the set $E(\mathcal{P})$.

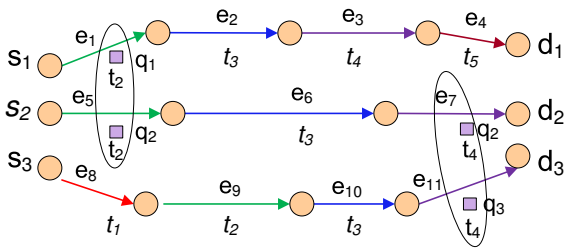


Figure 7.2: Illustrating the random delays technique: Paths for three source-destination pairs are shown. Let $C(e_5) = e_1, C(e_{11}) = e_7$. Each packet waits at its source for a delay chosen randomly and then moves one edge at a time on its respective path. Packet q_3 at source s_3 starts at time t_1, and packets q_1, q_2 at sources s_1, s_2 start at time t_2. Subsequently, packet q_1 reaches its destination d_1 at time t_5 and packets q_2, q_3 reach their respective destinations at time t_4. As a result of the random delays, packets on links e_1, e_5 are scheduled at time t_2 and packets on links e_7, e_{11} are scheduled at time t_4. These links interfere with each other, resulting in an invalid schedule.

7.5.3 End-to-end Scheduling

At this point, we have already chosen a path P_i for each connection (s_i, t_i), computed power levels on all the transmitting links, and computed the overall congestion $C = w_f$. We now need to schedule the packets along these chosen paths so that the overall latency is minimized and the schedule is conflict-free. Procedure Schedule is based on the **random delays** technique of Leighton, Maggs and Rao [31], and the techniques used to adapt random

Procedure Schedule

Input : V, K, Q, $\mathcal{P}$, E, J, C

Output: A valid schedule $\mathcal{S}$ that moves packets along their paths to their destinations

`//Random Delay`

foreach *packet* q_i *at source* s_i **do**
  Choose a delay $X_i \in \{1, .., C\}$ uniformly at random
end

`//Construct an invalid schedule` $\mathcal{S}'$

foreach *packet* q_i **do**
  wait at source s_i for time X_i before starting
  move one edge at a time along P_i after waiting X_i steps, i.e., move on the jth edge of P_i at step $X_i + j$
end

`//Convert` $\mathcal{S}'$ `into a valid schedule` $\mathcal{S}$

`//`$|\mathcal{S}'|$ `denotes the length (make-span) of` $\mathcal{S}'$

for $t = 1, 2, \ldots, |\mathcal{S}'|$ **do**
  `//Construct a partial schedule` $\mathcal{S}_t$
  E_t = set of links scheduled by $\mathcal{S}'$ in time t.
  $\mathcal{C}_t(e) = |C(e) \cap E_t|, \forall e \in E_t$
  `//Set partitioning`
  Partition E_t into sets $B = \{B_0, B_1, \ldots, B_{\lfloor \log \Delta \rfloor}\}$ such that $B_i = \{e \in E_t : 2^i \leq \ell(e) < 2^{i+1}\}$
  for $i = \lfloor \log \Delta \rfloor$ *down to* 0 **do**
    Partition links in set B_i into sets $W_i = \{W_0^i, W_1^i, ..., W_{\lfloor \log \Gamma_{node} \rfloor}^i\}$ such that $W_j^i = \{e = (u, v) \in B_i : 2^j \leq J(u) < 2^{j+1}\}$
    for $j = \lfloor \log \Gamma_{node} \rfloor$ *down to* 0 **do**
      `//Greedy Coloring`
      Order the edges in $E_t \cap W_j^i$ in non-increasing order of length $\{e_1, e_2, \ldots, e_s\}$
      for $k = 1$ *to* $|E_t \cap W_j^i|$ **do**
        Let $f(e_k)$ be a positive integer denoting the color that is to be assigned to edge e_k
        Assign e_k the smallest numbered color that is not used by any edge in $C(e_k) \cap \{e_1, \ldots, e_{k-1}\}$,i.e.,$f(e_k)$ is the smallest positive integer not in the set $\{f(e_{k'}) : k' < k,\ e_{k'} \in C(e_k)\}$.
      end
      Construct a schedule $\mathcal{S}_{t,i,j}$ for edges in $E_t \cap W_i^j$ by scheduling all edges e with color $f(e) = r$ in the rth step, for $r = 1, 2, \ldots$.
    end
    Construct schedule $\mathcal{S}_{t,i} = \mathcal{S}_{t,i,\log \Gamma_{node}} \bullet \ldots \bullet \mathcal{S}_{t,i,0}$ *(The* $\bullet$ *operator implies concatenation of different schedules)*
  end
  Construct schedule $\mathcal{S}_t = \mathcal{S}_{t,\log \Delta} \bullet \ldots \bullet \mathcal{S}_{t,0}$
end
Construct schedule $\mathcal{S} = \mathcal{S}_1 \bullet \ldots \bullet \mathcal{S}_\ell$, where $\ell = |\mathcal{S}'|$

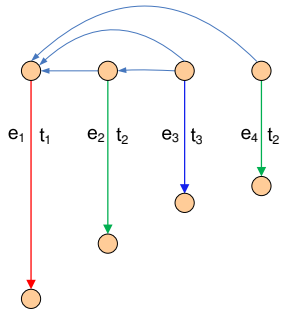


Figure 7.3: Illustrating the greedy coloring scheme: Let $C(e_2) = \{e_1\}, C(e_3) = \{e_1, e_2\}, C(e_4)\{e_1\}$. We first construct the interference graph. For a pair of edges $e_i = (u_i, v_i)$ and $e_j = (u_j, v_j)$, if $e_i \in C(e_j)$, then in the interference graph, we add a directed link from u_j to u_i. For example there is an edge from e_2 to e_1 as $e_1 \in C(e_2)$. We then consider the links in the decreasing order of their lengths and color each link such that no two interfering links have the same color assignment.

delays to geometric graphs in [29]. The congestion and dilation are important because we show that they are good lower bounds on the length of the optimum schedule, and it is possible to construct a schedule whose length is proportional to the congestion+dilation. The following is an informal description of the main steps in Procedure Schedule; see the procedure-description for the details.

1. Random Delays and an Invalid Schedule: At the outset of procedure Schedule, every packet waits at its source s_i for a delay X_i chosen randomly from $\{1.., C\}$, and then moves one edge at a time on the path $P_i \in \mathcal{P}$. This gives schedule $\mathcal{S}'$, which could be invalid because there could be simultaneous senders that violate the SINR constraints (cf. Figure 7.2).

2. Partial Schedule Construction: We break down the invalid schedule $\mathcal{S}'$ into different time steps t to obtain a partial schedule $\mathcal{S}'_t$. This schedule consists of all links E_t scheduled in $\mathcal{S}'$ at time t and we define partial congestion $\mathcal{C}_t(e)$ to be the congestion at link e at time t. We now convert the invalid schedule at each time step to a valid schedule by partitioning all the links into appropriate sets and then coloring all the links in the same set.

3. Partitioning: In this step we partition links in E_t into disjoint sets (bins) $B = \{B_0, B_1, \ldots, B_{\lfloor \log \Delta \rfloor}\}$ such that $B_i = \{e \in E_t : 2^i \leq \ell(e) < 2^{i+1}\}, i = 0, \ldots, \lfloor \log \Delta \rfloor$. We further partition each set B_i into a family of subsets $\{W_0^i, W_1^i, ..., W_{\lfloor \log \Gamma_{node} \rfloor}^i\}$ such that for $j = 1, \ldots, \lfloor \log \Gamma_{node} \rfloor$, $W_j^i = \{e = (u, v) \in B_i : 2^j \leq J(u) < 2^{j+1}\}$. The motivation behind partitioning links into different sets is to bound the number of nodes that can transmit simultaneously without violating the SINR constraints.

4. Greedy Coloring: After set partitioning, we color the links in each set W_j^i using a greedy coloring scheme. For any pair of links $e_l, e_m \in W_j^i$ such that either $e_l \in C(e_m)$ or $e_m \in C(e_l)$, the coloring scheme ensures that e_l, e_m are not assigned the same color (cf. Figure 7.3).

5. Combining different schedules: After partitioning and greedy coloring, we combine all the sub-schedules formed to get the final schedule $\mathcal{S}$.

Analysis of procedure Schedule

We show that the schedule obtained from the procedure Schedule is valid (conflict-free) and has a length that is at most a poly-log factor away from the optimal schedule.
The analysis of procedure Schedule consists of the following parts:

1. *Schedule validity*: We prove that the schedule produced by algorithm MinDelay is valid and the SINR constraints are satisfied at every time slot. Recall that we partition the links into different bins (sets) based on their edge lengths and power assignment and further schedule the links belonging to each bin. By utilizing a greedy scheduling scheme, we ensure that no two links e_i, e_j such that either $e_i \in C(e_j)$ or $e_j \in C(e_i)$ are assigned the same color. The binning strategy allows us to limit the number of links that can be scheduled simultaneously. Let $\mathcal{T}$ denote a set of simultaneously scheduled links at time t. We exploit the geometric nature of the problem, to obtain an upper bound on the total interference caused at any receiver v, where $e = (u, v) \in \mathcal{T}$ as a result of the transmissions on links $e' \in \mathcal{T}$. By selecting an appropriate value for

constant a, we deduce that the SINR at any receiver v is greater than the antenna gain β.

2. *Length of the schedule*: We first derive an upper bound on the length of the invalid schedule $\mathcal{S}'$ produced and show that it is $C + D$. Recall that at each time step t, we construct a partial schedule $\mathcal{S}_t$ by converting the invalid schedule into a valid schedule via a greedy coloring scheme. The length of the partial schedule $\mathcal{S}_t$ therefore depends on the congestion present at every link from set E_t, where E_t denotes the links scheduled at time t according to the invalid schedule $\mathcal{S}'$. We therefore estimate the expected congestion on any link from set E_t. By applying Chernoff bounds, we show that for any time t, the probability of congestion on any link from the set E_t being high is very low. The expected congestion estimate allows us to obtain an upper bound on the length of the partial schedule. By combining results from all the time slots, we obtain an upper bound on the length of the final schedule.

3. *Length of the optimal schedule*: In order to compare our results with the optimal schedule, we derive a lower bound on the length of the optimal schedule. As previously discussed, we develop a bi-criteria approximation algorithm and add a criteria that allows the optimal solution to choose power levels from the range $[j_{min}, (1-\epsilon)j_{max}]$. We first consider the set of links scheduled by the optimal solution at a given time t having similar power levels. By exploiting the geometric nature of the problem, we show that the length of this set is bounded and is a constant. Using this property, we obtain bounds on the length of the optimal schedule.

Validity of Schedule

Lemma 7.2. *The schedule $\mathcal{S}$ produced by procedure* Schedule *is valid in the sense that the SINR constraints are satisfied at all receivers at every step of the schedule, if $a^\alpha \geq 2^\alpha \frac{96\beta(\epsilon+1)}{\epsilon(\alpha-2)}$, $a \geq 2$, and $\alpha > 2$.*

Proof: We exploit the geometric nature of the problem to obtain an upper bound on the number of links that can be simultaneously scheduled. We then show that for every receiver v_k, the SINR at v_k due to simultaneous transmissions is always at least β.

Recall that the final schedule $\mathcal{S}$ is obtained by concatenating the partial schedules $\mathcal{S}_t$ in order; each partial schedule $\mathcal{S}_t$ is in turn constructed by putting together partial schedules $\mathcal{S}_{t,i}$, which are in turn constructed by putting together partial schedules $\mathcal{S}_{t,i,j}$ for edges in $E_t \cap W_j^i$. Therefore, it suffices to prove that each partial schedule $\mathcal{S}_{t,i,j}$ is valid, which we argue next.

Consider any step t' of schedule $\mathcal{S}_{t,i,j}$. Let $\mathcal{T} = \{e_k = (u_k, v_k) : k = 1, \ldots, \tau\}$ denote the set of edges that are simultaneously scheduled at step t' of $\mathcal{S}_{t,i,j}$. In order to show that $\mathcal{S}_{t,i,j}$ is valid, we have to prove that all the transmissions in $\mathcal{T}$ happen successfully, i.e., the SINR at each v_k for $e_k = (u_k, v_k) \in \mathcal{T}$ is at least β; we will show here that the SINR at any receiver v_1 such that $e_1 = (u_1, v_1) \in \mathcal{T}$ is at least β.

We will first show that disks of radius $\frac{a}{4}\ell(e_1)$ (where $\ell(e_1)$ denotes the length of edge e_1) centered at each u_k, where $e_k = (u_k, v_k) \in \mathcal{T}$ are disjoint. By construction, all edges in $\mathcal{T}$ have the same color (i.e., have been assigned same time slot for transmission). Therefore, we must have $e_k \notin C(e_{k'})$, for any distinct $e_k, e_{k'} \in \mathcal{T}$ (otherwise, the coloring step would have assigned different colors to these two edges). This means that $d(u_k, u_{k'}) > a \max\{\ell(e_k), \ell(e_{k'})\}$. Further due to the partitioning (steps 11 through 14) we have $\mathcal{T} \subseteq (B_i \cap W_i^j)$. Therefore, $\ell(e_k) \in [2^i, 2^{i+1})$ for each $e_k \in \mathcal{T}$ and $\forall e_k = (u_k, v_k) \in \mathcal{T}$, $J(u_k) \in [2^j, 2^{j+1})$. This implies $d(u_k, u_{k'}) > a2^i \geq \frac{a}{2}\ell(e_1)$ for any distinct $e_k, e_{k'} \in \mathcal{T}$. Therefore, disks of radius $\frac{a}{4}\ell(e_1)$ centered at each u_k where $e_k = (u_k, v_k) \in \mathcal{T}$ are disjoint.

We will now calculate the $SINR(v_1)$ due to all the other transmissions in $\mathcal{T}$. As in [42], we will partition the plane into rings centered at u_1, in order to compute the interference at v_1. Consider rings R_m, $m = 0, 1, \ldots$ of width $a\ell(e_1)$ around u_1. R_m contains all links $e_k = (u_k, v_k) \in \mathcal{T}$ for which $ma\ell(e_1) \leq d(u_1, u_k) < (m+1)a\ell(e_1)$ (cf. Figure 7.5.3). We know that any given link $e_k = (u_k, v_k) \in \mathcal{T}$, does not interfere with link $e_1 = (u_1, v_1)$. Therefore

$\forall e_k \in \mathcal{T}, e_k \neq e$, we have $d(u_1, u_k) > a \max\{\ell(e_1), \ell(e_k)\}$. Therefore the first ring R_0 will not contain any other links from set $\mathcal{T}$, except for link e_1. The area of the ring R_m can be calculated as,

$$\begin{aligned} A(R_m) &= \pi[((m+1)a\ell(e_1))^2 - (ma\ell(e_1))^2] \\ &= \pi a^2(2m+1)\ell(e_1)^2 \\ &\leq 3\pi m a^2 \ell(e_1)^2. \end{aligned}$$

Next, the non-overlapping disks property above also implies that the number of nodes transmitting in R_m for $m \geq 1$ is at most

$$\frac{3\pi m a^2 \ell(e_1)^2}{\pi a^2 \ell(e_1)^2/16} = 48m.$$

Also, for $m \geq 1$, for each $e_k \in \mathcal{T} \cap R_m$, we have $d(u_k, v_1) \geq (am-1)\ell(e_1)$. Since we have $a \geq 2$, we have $d(u_k, v_1) \geq (am-1)\ell(e_1) \geq \frac{am}{2}\ell(e_1)$. Therefore, the interference at v_1 due to nodes in R_m, denoted by $\mathcal{I}_m(v_1)$, can be upper bounded as follows: since $J(u_k) \leq 2J(u_1)$ for all k we have,

$$\begin{aligned} \mathcal{I}_m(v_1) &\leq 48m 2^\alpha \frac{2J(u_1)}{(am\ell(e_1))^\alpha} \\ &= 2^\alpha \frac{96J(u_1)}{a^\alpha m^{\alpha-1}\ell(e_1)^\alpha}. \end{aligned}$$

Summing up the interference over all rings R_m, we have,

$$\begin{aligned} \sum_{m=1}^{\infty} \mathcal{I}_m(v_1) &\leq 2^\alpha \frac{96J(u_1)}{a^\alpha \ell(e_1)^\alpha} \sum_{m=1}^{\infty} \frac{1}{m^{\alpha-1}} \\ &\leq 2^\alpha \frac{96J(u_1)}{a^\alpha \ell(e_1)^\alpha} \int_1^\infty \frac{dx}{x^{\alpha-1}} \\ &\leq 2^\alpha \frac{96J(u_1)}{a^\alpha \ell(e_1)^\alpha (\alpha-2)}. \end{aligned}$$

Therefore the SINR at receiver v_1 is

$$\begin{aligned} SINR(v_1) &\geq \frac{J(u_1)}{\ell(e_1)^\alpha[N + 2^\alpha \frac{96J(u_1)}{a^\alpha \ell(e_1)^\alpha(\alpha-2)}]} \\ &\geq \frac{J(u_1)}{\ell(e_1)^\alpha[N + 2^\alpha \frac{96J(u_1)}{\ell(e_1)^\alpha}\frac{\epsilon}{96\beta(1+\epsilon)}]} \\ &= \frac{J(u_1)}{\ell(e_1)^\alpha[N + 2^\alpha \frac{\epsilon J(u_1)}{(1+\epsilon)\beta\ell(e_1)^\alpha}]} \end{aligned}$$

where the second inequality above follows from the condition that $a^\alpha \geq 4^\alpha \frac{48\beta(\epsilon+1)}{\epsilon(\alpha-2)}$ (cf. Chapter 2, Section 2.5.4). The last expression above is at least β if $J(u_1) \geq (1+\epsilon)\beta N \ell(e_1)^\alpha$, which is ensured in the path selection and power control stage. Therefore, the schedule $\mathcal{S}$ produced by procedure Schedule is valid.

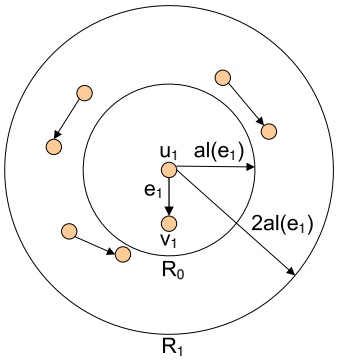


Figure 7.4: For a given link $e_1 = (u_1, v_1)$, construct rings of radius $a\ell(e_1)$ around u_1. We calculate the interference experienced by node v_1 due to other simultaneously transmitting links.

Length of the schedule

Lemma 7.3. $|\mathcal{S}'| \leq C + D$.

Proof: Packet q_i reaches its destination at step $X_i + |\mathcal{P}_i|$. Since $X_i \leq C$, $|\mathcal{S}'| \leq C + D$.

Lemma 7.4. *The length of the partial schedule $\mathcal{S}_t$ produced by procedure* Schedule *in step 9 is $O(\max_e |C(e) \cap E_t| \log \Delta \log \Gamma_{node})$.*

Proof: Procedure Schedule schedules the edges in E_t by putting together schedules $\mathcal{S}_{t,i}$, $i = \log \Delta, \ldots, 0$,which in turn is formed by putting together schedules $\mathcal{S}_{t,i,j}$, $j = \log \Gamma_{node}, \ldots, 0$ which forms a schedule of the edges in $E_t \cap B_i \cap W_i^j$. We will argue below that for each $j = 0, \ldots, \log \Gamma_{node}$, the edges in $E_t \cap B_i \cap W_i^j$ can be scheduled in time $O(\max_e |C(e) \cap E_t \cap B_i|)$; the Lemma then follows. Consider the coloring process in steps 15 through 19 of the algorithm. Let $E_t \cap B_i \cap W_i^j = \{e_1, \ldots, e_s\}$, with $\ell(e_1) \geq \ell(e_2) \geq \ldots \geq \ell(e_s)$. The algorithm chooses a color for each e_k in this order - the color chosen for e_k is the smallest numbered color that is not used by $C(e_k) \cap \{e_1, \ldots, e_{k-1}\}$. Therefore, the number of colors used is at most $\max_e |C(e) \cap E_t \cap B_i| + 1$. Since there are $\log \Gamma_{node}$ such sets and $\log \Delta$ sets for each of the $\log \Gamma_{node}$ sets, the lemma follows.

Lemma 7.5. *For each $t = 1, \ldots, |\mathcal{S}'|$,*

$$Pr[\max_e \mathcal{C}_t(e) \geq 24 \log \max\{n, C + D\}] \leq \frac{1}{n(C + D)}$$

where $\mathcal{C}_t(e) = |C(e) \cap E_t|$.

Proof: Fix any time t. For each edge $e \in E$, we define $Q(e) = \{q_i : P_i \cap C(e) \neq \phi\}$ to be the set of packets passing through some edge of $C(e)$. Consider any edge e. Let Y_i be an indicator variable that is 1 if packet $q_i \in Q(e)$ crosses some $e' \in C(e)$ at time t in $\mathcal{S}'$. Then, $\mathcal{C}_t(e) = |C(e) \cap E_t| = \sum_{q_i \in Q(e)} Y_i$. As in the proof of Theorem 7.1, let $n(e, P_i) = |C(e) \cap P_i|$. If $e' \in C(e)$ is the ℓth edge in P_i, packet q_i would be on edge e' at time t in schedule $\mathcal{S}'$ if and only if $X_i = t - \ell$; the probability of this event is at most $1/C$. Therefore, $\Pr[Y_i = 1] \leq n(e, P_i)/C$, which implies

$$\begin{aligned} E[\mathcal{C}_t(e)] &= \sum_{q_i \in Q(e)} \Pr[Y_i = 1] \\ &\leq \sum_{q_i \in Q(e)} n(e, P_i)/C \\ &= \frac{1}{C} \sum_{P_i : P_i \cap C(e) \neq \phi} n(e, P_i) \\ &\leq 6 \log n, \end{aligned}$$

where the last inequality follows from Theorem 7.1. The variables Y_i are independent Bernoulli variables, and so the Chernoff bound can be applied to the sum $\mathcal{C}_t(e)$, which gives

$$\Pr[\mathcal{C}_t(e) \geq 24 \log \max\{n, C + D\}] \leq \frac{1}{n^3(C + D)}$$

By a union bound, it follows that

$$Pr[\max_e \mathcal{C}_t(e) \geq 24 \log \max\{n, C + D\}] \leq \frac{1}{n(C + D)}.$$

Lemma 7.6. *$|\mathcal{S}|$ is $O((C + D) \log n \log \Delta \log \Gamma_{node})$, with probability at least $1 - \frac{1}{n}$.*

Proof: Applying the union bound along with Lemma 7.5 and Lemma 7.3 we get

$$Pr[\exists\ t :\ \max_e \mathcal{C}_t(e) \geq 24 \log \max\{n, C + D\}] \leq \frac{C + D}{n(C + D)},$$

i.e., at most $1/n$. Thus, $|\mathcal{S}_t| \leq 24 \log \max\{n, C + D\} \log \Delta \log \Gamma_{node}$ for all t, with probability at least $1 - \frac{1}{n}$, which implies that the length of $\mathcal{S}$ is at most $O((C + D) \log n \log \Delta \log \Gamma_{node})$ with this probability.

Note that the bound on the schedule length holds, irrespective of the constraints on the constant a, and on the power level. It is for the validity of the schedule that we need these additional constraints.

Length of the Optimal Schedule

We now derive a lower bound on the length of the optimal schedule assuming that the power levels are chosen from the range $[j_{min}, (1-\epsilon)j_{max}]$.

Lemma 7.7. *Consider any schedule $\mathcal{S}_{OPT}(j_{min}, (1-\epsilon)j_{max})$ of optimal length for the given problem, that uses power levels from the range $[j_{min}, (1-\epsilon)j_{max}]$. Then, $|\mathcal{S}_{OPT}(j_{min}, (1-\epsilon)j_{max})| = \Omega((C_{OPT} + D_{OPT})/\log \Gamma_{node})$, where $C_{OPT} + D_{OPT}$ denotes the smallest congestion plus dilation that is possible if all paths have links $e = (u, v)$ such that $j_{max}(1-\epsilon) \geq \beta N d(u,v)^{\alpha}$ for some constant ϵ.*

Proof: Consider an optimal set of paths $\mathcal{P}_{OPT}$ that has congestion and dilation equal to C_{OPT} and D_{OPT}, respectively. Let $G_i = \{e_k = (u_k, v_k) \in E : 2^i \leq J(u_k) < 2^{i+1}\}$, for $i = 1, \ldots, \log \Gamma_{node}$. We now fix any $e = (u, v)$ and any time step t. Let $A_t = \{e_j = (u_j, v_j) : j = 1, \ldots, r\}$ be the set of links in $C(e)$ that are scheduled in $\mathcal{S}_{OPT}(j_{min}, (1-\epsilon)j_{max})$ at time t. Further let $H_{t,i} = \{e_k \in A_t \cap G_i\}$. We first argue that the number of links that can be simultaneously scheduled from the set $H_{t,i}$ for any edge e, at any time t and any $i \in \{1, \ldots, \log \Gamma_{node}\}$ is $O(1)$. Let the links in set $H_{t,i}$, be arranged in the non-decreasing order of their lengths. Let there be s links in set $H_{t,i}$. Therefore $d(u_1, v_1) \leq d(u_2, v_2) \leq ... \leq d(u_s, v_s)$. Since all these links are used simultaneously, the SINR at each node v_j should exceed β. We shall compute the SINR at receiver v_s.

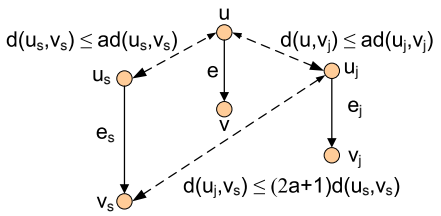


Figure 7.5: For a given link $e = (u, v)$ and set of other links $H_{t,i} = \{e_s, e_j\} \in C(e) \cap G_i$ scheduled by $\mathcal{S}_{OPT}(j_{min}, (1-\epsilon)j_{max})$ at time t, $d(u_j, v_s) \leq (2a+1)d(u_s, v_s)$

Consider a link $e_j = (u_j, v_j) \in H_{t,i}$ (refer to figure 7.5.3). Since link $e_j, e_s \in C(e)$, according to the definition of $C(e)$, we know that $d(u, u_j) \leq ad(u_j, v_j)$ and $d(u, u_s) \leq ad(u_s, v_s)$. Further since $e_j = (u_j, v_j), e_s = (u_s, v_s) \in G_i$, we have $\frac{J(u_s)}{2} \leq J(u_j) \leq 2J(u_s)$.
It can be seen that

$$\begin{aligned} d(u_j, v_s) &\leq d(u_j, u_s) + d(u_s, v_s) \\ &\leq d(u, u_j) + d(u, u_s) + d(u_s, v_s) \\ &\leq (2a+1)d(u_s, v_s). \end{aligned}$$

The interference experienced by v_s due to all such e_j is

$$I_r(v_s) = \sum_{e_j=(u_j,v_j)\in H_{t,i}, j\neq s} \frac{J(u_j)}{d(u_j, v_s)^\alpha}$$

Therefore in order to satisfy the SINR constraint at node v_s, we need

$$\begin{aligned} \frac{J(u_s)}{d(u_s, v_s)^\alpha[N + I_r(v_s)]} &\geq \beta \\ \implies J(u_s) \geq \beta d(u_s, v_s)^\alpha[N &+ I_r(v_s)] \\ \implies J(u_s) \geq \beta d(u_s, v_s)^\alpha[N &+ \sum_{e_j=(u_j,v_j)\in H_{t,i}, j\neq s} \frac{J(u_j)}{d(u_j, v_s)^\alpha}] \\ \implies J(u_s) \geq \beta d(u_s, v_s)^\alpha[&(s-1)\frac{J(u_s)}{2((2a+1)d(u_s, v_s))^\alpha}] \end{aligned}$$

The above condition is satisfied for $s \leq 2\frac{(2a+1)^\alpha}{\beta} + 1$, which is a constant. Therefore for any e, t and for any $i \in \{1, \ldots, \log \Gamma_{node}\}$, the number of links $e' \in C(e) \cap H_{t,i}$ that can be simultaneously scheduled in $\mathcal{S}_{OPT}(j_{min}, (1-\epsilon)j_{max})$ is $O(1)$.

Next, we argue that $|\mathcal{S}_{OPT}(j_{min}, (1-\epsilon)j_{max})| = \Omega(C/\log \Gamma_{node})$. It can be seen that, for a given edge e, some $i \in \{1, \ldots, \log \Gamma_{node}\}, |C(e) \cap G_i| \geq |C(e)|/\log \Gamma_{node}$. Let n_t be the number of edges from $C(e) \cap G_i$ scheduled in $\mathcal{S}_{OPT}(j_{min}, (1-\epsilon)j_{max})$ at time t. Then, $n_1 + n_2 + ... + n_{|\mathcal{S}_{OPT}(j_{min},(1-\epsilon)j_{max})|} = |C(e) \cap G_i| \geq |C(e)|/\log \Gamma_{node}$. From our discussion above, $n_t = O(1)$, for each t. This implies that $|\mathcal{S}_{OPT}(j_{min}, (1-\epsilon)j_{max})| = \Omega(C_{OPT}/\log \Gamma_{node})$.

7.5.4 Analysis of the Algorithm MinDelay

We show that if the power levels for every node are chosen from the range $[j_{min}, j_{max}]$, algorithm MinDelay gives a poly-log-factor bi-criteria approximation for the end-to-end latency:

Theorem 7.2. *Algorithm* MinDelay *yields end-to-end latency of at most* $O(\log^2 n \log^3 \Delta \cdot \mathcal{S}_{\mathcal{OPT}}(j_{min}, (1-\epsilon)j_{max}))$*, where* $\mathcal{S}_{\mathcal{OPT}}(j_{min}, (1-\epsilon)j_{max})$ *denotes the optimal latency of minimum length possible if the power levels are chosen from the range* $[j_{min}, (1-\epsilon)j_{max}]$*, for any given parameter* $\epsilon > 0$*.*

Proof: From Lemma 7.6, we know that the maximum length of the schedule produced is $O((C+D)\log n \log \Delta \log \Gamma_{node})$,from Theorem 7.1 we have $C+D = O((C_{OPT}+D_{OPT})\log n)$ and from Lemma 7.7 we have $|\mathcal{S}_{OPT}(j_{min}, (1-\epsilon)j_{max})| = \Omega((C_{OPT}+D_{OPT})/\log \Gamma_{node})$. Therefore by combining the results of Lemma 7.6 and 7.7 and observing that $\log \Gamma_{node} = O(\log \Delta)$ we have end-to-end latency of at most $O(\log^2 n \log^3 \Delta \mathcal{S}_{\mathcal{OPT}}(j_{min}, (1-\epsilon)j_{max}))$.

The bounds achieved for algorithm MinDelay are worst-case approximation bounds. The algorithm complexity depends on (i) number of nodes (n) in the network and (ii) constants Δ, Γ_{node}. As mentioned earlier, δ denotes the maximum inter-point separation. and Γ_{node} denotes the ratio between maximum and minimum power assigned to nodes belonging to set V. It should be noted that the current implementation of algorithm MinDelay is best suited for moderately dense and closely confined ad hoc networks.

7.6 Corollaries and Extensions

We now sketch techniques for solving variations/extensions of our CLM problem.

1. *End-to-end delay minimization with joint routing and scheduling for fixed power levels*: This is a variation of the CLM problem wherein the power levels are fixed. Algorithm MinDelay could be used to solve this problem, with a few modifications. The constraints

(7.6), which ensure that power assigned is valid, need to be removed from the LP formulation in procedure PathSelection. Procedure PowerControl is not required.

2. *End-to-end delay minimization for fixed routes and power levels*: This is again a variation of CLM; it can be solved by directly using procedure MinDelay. Other procedures are not required.

3. *End-to-end delay minimization with joint routing, scheduling and power control with bound on total energy consumed*: This is an extension of CLM, where we are given a bound B on the total energy consumed. This can be solved by adding the constraint $\sum_i \sum_{e=(u,v)\in E} y(i,e)(1+\epsilon)\beta N d(u,v)^\alpha \leq B$ to the LP formulation in procedure PathSelection.

7.7 Most Relevant Prior Work

Recent work by Moscibroda *et al.* [40] studies the problem of scheduling packets with SINR constraints to ensure strong-connectedness of the graph. This is a topology-control problem that aims at finding a schedule of minimal length in which all transmitting links form a strongly-connected network. The authors study the scheduling complexity of connecting a given number of nodes located at arbitrary positions by a communication tree and present a scheduling algorithm that successfully schedules a strongly connected set of links in poly-logarithmic time. We consider a slightly different variation of the problem considered in [40]; in our CLM problem, for a given source-destination pairs, we initially find paths, and then schedule packets on the selected paths. Subsequent work by Moscibroda *et al.* [42], studies the MAC level scheduling problem with SINR constraints. The authors present a scheduling algorithm that successfully schedules a set of links in poly-logarithmic time by assigning non-uniform power to the transmitting nodes. The results presented in [42] are not compared with the optimal solution; it is therefore difficult to gauge the quality of the solutions. Further, extending the techniques for deriving the end-to-end scheduling complexity is not

intuitive. Motivated by the work and techniques of [42], our CLM problem attempts to provide approximation bounds that are comparable with the optimal solution for the end-to-end scheduling problem. The work by Kumar *et al.* [29] derives poly-logarithmic approximation bounds for minimizing end-to-end latencies. They develop distributed protocols for routing and scheduling for various families of disk graphs (that capture the geometric nature of the wireless interference). In our work, we extend some of the techniques from [29] for solving the CLM problem for more difficult and realistic SINR interference model. In a recent work by Brar *et al.* [6], the authors present a polynomial time efficient heuristic for computing a feasible schedule under the SINR interference model and derive an approximation factor for the length of the schedule for random distribution of nodes. In our CLM problem, we provide a approximation factor for any arbitrary node distribution.

7.8 Summary

We described a general algorithmic technique that lead to efficient polynomial-time centralized approximation algorithms for minimizing end-to-end latency by jointly considering routing, scheduling and power control layers. The results extend recent work of [29, 40, 42] by simultaneously considering multiple layers in the stack and more realistic models of radio interference.

Chapter 8

Conclusions

8.1 Summary of Contributions

In this dissertation, we studied various algorithmic aspects of capacity in multi-hop wireless networks. We developed innovative approximation algorithms by combining various powerful techniques for estimating the capacity of any given multi-hop wireless network. Our techniques incorporate cross-layer constraints and are generic enough to accommodate different performance objectives and constraints. We summarize our contributions as follows:

- We compared the SINR and the graph-based models for the same instance with the same power levels and observed that the predicted throughput can be significantly different in these two models. Our results suggest that both these interference models have their own limitations and more research is still required in developing realistic interference models.

- We theoretically studied the problem of maximizing the throughput of any given wireless network with SINR constraints and developed a provable approximation algorithm for this problem by means of a linear programming formulation. We conducted extensive simulations to validate and evaluate the performance of our approximation

algorithms.

- We studied the problem of joint optimization of the total throughput rate and the total power consumed in a given multi-hop wireless network with adaptive power control. Our central contribution here was a general algorithmic technique that lead to an efficient polynomial time approximation algorithm by considering dynamic power assignments, link rate changes and changes in the interference graph.

- We considered the problem of minimizing latency (length of the end-to-end schedule) in wireless multi-hop networks. We described a general algorithmic technique leading to efficient polynomial-time randomized approximation algorithms for minimizing end-to-end latency by jointly considering routing, scheduling and power control layers.

8.2 Future Directions

We now survey some specific open problems as well as some broad directions for future research that are inspired by this dissertation.

- Distributed Algorithms: In Chapters 4, 6, and 7 we developed algorithms that require centralized control. A natural direction would be to extend these techniques to a distributed setting. The work of [18] provides techniques for solving linear programs in a distributed fashion. These techniques could be applied to linear programs developed in this work. As a next step towards developing fully distributed algorithms, one would also need to develop an efficient distributed scheduling scheme. Recent work by [9] has developed a distributed random access scheduling strategy for wireless networks that yields a rate vector that is provably competitive with respect to the optimal centralized scheduling strategy. The authors develop an approximation technique to approximate the non-convex feasible link-rate region and compute the channel access probability for every link. An open research problem would be to investigate if the approximation

techniques discussed in [9] can be extended to develop a distributed scheduling scheme for the TM-SINR and the PETM problems discussed in Chapters 4 and 6.

- Performance bounds: We developed poly-log factor approximation bounds for the TM-SINR and the CLM problems in Chapter 4 and 7. One can analyze if these bounds can be further improved. In particular, is it possible to derive constant-factor approximation bounds for these problems? From our experimental analysis in Chapter 5 we discovered that the congestion measure proposed in this work could lead to pessimistic estimates. Can this congestion measure be improved to provide better capacity estimates?

- Power control with SINR constraints: In this work we have not considered the problem of maximizing throughput by performing adaptive power control with the SINR interference model. One can extend some of our results and techniques for the TM-SINR and PETM problem (cf. Chapter 4, 6) to solve this problem.

- Capacity of hybrid networks: A hybrid network is formed by placing base stations or access points (referred as hybrid nodes) in an ad-hoc network. The hybrid nodes are connected to each other via wired links and they act as relays for other wireless nodes. Hybrid networks can be viewed as a compromise between high cost, high performance infrastructure-based networks, and low-maintenance, low performance ad- hoc networks. Recent work by [38] attempts to mathematically characterize the capacity of these hybrid networks for random topologies. An open research problem would be to investigate the capacity of any given instance of a hybrid network. Can the techniques proposed in this work be utilized for solving this problem?

- Network Planning: Current work in wireless capacity focuses mainly on the design of wireless communication protocols which are guaranteed to achieve the capacity of the network. A direction for future research is to jointly design the wireless network and the communication protocols. How can we efficiently design a wireless network so that

it utilizes its maximum capacity? How can we determine the position of the wireless nodes and access-points?

- Scheduling in Cognitive Networks: Recently there has been significant interest in developing cognitive networks and radios. The idea is to construct intelligent wireless networks that can adapt to user needs and changing environmental conditions. Dynamic spectrum allocation is an important aspect in cognitive networks in which every transmission node could have channels dynamically allocated and de-allocated based on the user demands. Wireless interference becomes more prominent in these networks, and hence the use of physical interference model in protocol design is important. Constructing schedules for allocating the time-spectrum blocks with the physical interference model in order to optimize the overall network performance is an open research problem. One can investigate if the scheduling schemes developed in this work can be extended for solving the scheduling problem in cognitive networks.

Bibliography

[1] A. Agarwal and P. Kumar. Capacity bounds for ad hoc and hybrid wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(3):71–81, July 2004.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38:393–422, 2002.

[3] M. Alicherry, R. Bhatia, and E. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proc. ACM MOBICOM*, pages 58–72, August 2005.

[4] C. Barrett, A. Marathe, M. Marathe, and D. Martin. Characterizing the interaction between routing and MAC protocols in ad-hoc networks. In *Proc. ACM MOBIHOC*, pages 92–103, 2002.

[5] R. Bhatia and M. Kodialam. On power efficient communication over multi-hop wireless networks: Joint routing, scheduling and power control. In *Proc. IEEE INFOCOM*, volume 2, pages 1457–1466, March 2004.

[6] G. Brar, D. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *Proc. ACM MOBICOM*, pages 2–13, 2006.

[7] C. Buragohain, S. Suri, C. Toth, and Y. Zhou. Improved throughput bounds for interference-aware routing in wireless networks. In *Proc. International Computing and Combinatorics Conference*, July 2007.

[8] D. Chafekar, V.S. Anil Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Cross-layer latency minimization in wireless networks with SINR constraints. In *Proc. ACM MOBIHOC*, pages 110–119, September 2007.

[9] D. Chafekar, D. Levin, V.S. Anil Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Capacity of asynchronous random-access scheduling in wireless networks. In *Proc. IEEE INFOCOM*, pages 1148–1156, April 2008.

[10] P. Chaporkar, K. Kar, and S. Sarkar. Achieving queue-length stability through maximal scheduling in wireless networks. In *Proc. Information Theory and Applications Inaugural Workshop*, pages 6–10, 2005.

[11] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *Proc. Allerton Conference on Communication, Control and Computing*, pages 28–30, 2005.

[12] L. Chen, S. Low, M. Chiang, and J. Doyle. Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks. In *Proc. IEEE INFOCOM*, pages 1–13, April 2006.

[13] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[14] IEEE Computer Society LAN MAN Standards Committee. Wireless lan medium access control (MAC) and physical layer (PHY) specifications. In *IEEE Std. 802.11*, 1997.

[15] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[16] CPLEX. http://www.ilog.com/products/cplex/.

[17] R. L. Cruz and A. V. Santhanam. Optimal routing, link scheduling and power control in multi-hop wireless networks. In *Proc. IEEE INFOCOM*, pages 702–711, March 2003.

[18] N. Garg and J. Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *IEEE Symposium on Foundations of Computer Science*, pages 300–309, 1998.

[19] O. Goussevskaia, Y. Oswald, and R. Wattenhofer. Complexity in geometric SINR. In *Proc. ACM MOBIHOC*, pages 100–109, September 2007.

[20] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *Proc. IEEE INFOCOM*, volume 10, pages 477–486, April 2001.

[21] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[22] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *Proc. ACM MOBICOM*, pages 66–80, September 2003.

[23] C. Joo and N. Shroff. Performance of random access scheduling schemes in multi-hop wireless networks. In *Proc. IEEE INFOCOM*, pages 19–27, May 2007.

[24] J. Kleinberg. Approximation algorithms for disjoint paths problems. In *PhD. dissertation*, 1996.

[25] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless networks: The joint routing and scheduling problem. In *Proc. ACM MOBICOM*, pages 42–54, September 2003.

[26] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless networks wireless mesh networks with orthogonal channels. *IEEE/ACM Transactions on Networking (TON)*, pages 868–880, August 2005.

[27] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *Proc. ACM MOBICOM*, pages 73–87, August 2005.

[28] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Unit disk graph approximation. In *Proc. ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, 2004.

[29] V.S. Anil Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end packet-scheduling in wireless ad-hoc networks. In *Proc. SODA*, pages 1021–1030, 2004.

[30] V.S. Anil Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Algorithmic aspects of capacity in wireless networks. In *Proc. ACM SIGMETRICS*, pages 133–144, June 2005.

[31] T. Leighton, B. Maggs, and S.Rao. Packet routing and job shop scheduling in O(congestion+dilation) steps. *Combinatorica*, 14(2):167–180, 1994.

[32] F. Li, M. Li, R. Lu, H. Wu, M. Claypool, and R. Kinicki. Tools and techniques for measurement of ieee 802.11 wireless networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–8, April 2006.

[33] J. Li, C. Blake, D. Couto, H. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proc. ACM MOBICOM*, pages 61–69, July 2001.

[34] X. Lin and S. Rasool. Constant-time distributed scheduling policies for ad hoc wireless networks. In *Proc. IEEE Conference on Decision and Control*, pages 1258–1263, 2006.

[35] X. Lin and N. Shroff. Joint rate control and scheduling in multihop wireless networks. *43rd IEEE Conference on Decision and Control*, 2:1484–1489, December 2004.

[36] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks. In *Proc. IEEE INFOCOM*, volume 3, pages 1804–1814, March 2005.

[37] X. Lin, N. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications on Non-Linear Optimization of Communication Systems*, 24(8):1452–1463, June 2006.

[38] B. Liu, Z. Liu, and D.Towsley. On the capacity of hybrid wireless networks. In *Proc. IEEE INFOCOM*, volume 2, pages 1543–1552, March 2003.

[39] M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, 2005.

[40] T. Moscibroda and R. Wattenhofer. The complexity of connectivity in wireless networks. In *Proc. IEEE INFOCOM*, pages 1–13, April 2006.

[41] T. Moscibroda, R. Wattenhofer, and A. Zollinger. Protocol design beyond graph-based models. In *Proc. ACM SIGCOMM Workshop on HotNets*, November 2006.

[42] T. Moscibroda, R. Wattenhofer, and A. Zollinger. Topology control meets SINR: The scheduling complexity of arbitrary topologies. In *Proc. ACM MOBIHOC*, pages 310–321, May 2006.

[43] M. Neely. Optimal energy and delay tradeoffs for multi-user wireless downlinks. In *Proc. IEEE INFOCOM*, pages 1–13, April 2006.

[44] S. Parthasarathy. Resource allocation in networked and distributed environments. In *PhD. dissertation*, July 2006.

[45] C. Perez-Vega, J. Garcia, and J. Higuera. A simple and efficient model for indoor path-loss prediction. *Measurement science and technology*, 8:1166–1173, 1997.

[46] L. Qiu, Y. Zhang, F. Wang, M. Han, and R. Mahajan. A general model of wireless interference. In *Proc. ACM MOBICOM*, pages 171–182, 2007.

[47] Qualnet. http://www.scalable-networks.com/.

[48] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *Journal of Computer and System Sciences*, 37(2):130–143, October 1988.

[49] P. Raghavan and S. Thompson. Randomized rounding:a technique for provable good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[50] S. Ramanathan. A unified framework and algorithm for (T/F/C) dma channel assignment in wireless networks. In *Proc. IEEE INFOCOM*, pages 900–907, April 1997.

[51] S. Ramanathan and E. Lloyd. Scheduling algorithms for multi-hop radio networks. *IEEE/ACM Transactions on Networking*, 1:166–172, 1993.

[52] Neos Solvers. http://neos.mcs.anl.gov/neos/solvers/index.html.

[53] A. Srinivasan. Approximation algorithms via randomized rounding: A survey. In *Series in Advanced Topics in Mathematics, Polish Scientific Publishers PWN*, pages 9–71, 1999.

[54] A. Srinivasan and C. Teo. A constant factor approximation algorithm for packet routing and balancing local vs. global criteria. *SIAM Journal on Computing*, 30:2051–2068, 2001.

[55] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transacations on Automatic Control*, 4:2130–2132, December 1990.

[56] TRANSIMS. http://transims.tsasa.lanl.gov.

[57] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.

[58] S. Yi, Y. Pei, and S. Kalyanaraman. On the capacity improvement of ad hoc wireless networks using directional antennas. In *Proc. ACM MOBIHOC*, pages 108–116, 2003.

[59] H. Zimmermann. OSI reference model–The ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications*, 28(4):425–432, 1980.