Distributed Scheduling and Delay-Throughput Optimization in Wireless Networks under the Physical Interference Model

Guanhong Pei

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

Electrical Engineering

Anil Vullikanti, Chair

Y. Thomas Hou

Madhav V. Marathe

Binoy Ravindran

Aravind Srinivasan

October 12th, 2012
Blacksburg, Virginia

Keywords: Wireless Networks, Cross-layer Design, Physical Interference, Approximation Algorithms, Distributed Algorithms

Distributed Scheduling and Delay-Throughput Optimization in Wireless Networks under the Physical Interference Model

Guanhong Pei

ABSTRACT

We investigate diverse aspects of the performance of wireless networks, including throughput, delay and distributed complexity. One of the main challenges for optimizing them arises from radio interference, an inherent factor in wireless networks. Graph-based interference models represent a large class of interference models widely used for the study of wireless networks, and suffer from the weakness of over-simplifying the interference caused by wireless signals in a local and binary way. A more sophisticated interference model, the physical interference model, based on SINR constraints, is considered more realistic but is more challenging to study (because of its non-linear form and non-local property). In this dissertation, we study the connections between the two types of interference models — graph-based and physical interference models — and tackle a set of fundamental problems under the physical interference model; previously, some of the problems were still open even under the graph-based interference model, and to those we have provided solutions under both types of interference models.

The underlying interference models affect scheduling and power control — essential building blocks in the operation of wireless networks — that directly deal with the wireless medium; the physical interference model (compared to graph-based interference model) compounds the problem of efficient scheduling and power control by making it non-local and non-linear. The system performance optimization and tradeoffs with respect to throughput and delay require a "global" view across transport, network, media access control (MAC), physical layers (referred to as cross-layer optimization) to take advantage of the control planes in different levels of the wireless network protocol stack. This can be achieved by regulating traffic rates, finding traffic flow paths for end-to-end sessions, controlling the access to the wireless medium (or channels), assigning the transmission power, and handling signal reception under interference.

The theme of the dissertation is distributed algorithms and optimization of QoS objectives under the physical interference model. We start by developing the *first* low-complexity distributed scheduling and power control algorithms for maximizing the efficiency ratio for different interference models; we derive end-to-end per-flow delay upper-bounds for our sched-

uling algorithms and our delay upper-bounds are the *first* network-size-independent result known for multihop traffic. Based on that, we design the *first* cross-layer multi-commodity optimization frameworks for delay-constrained throughput maximization by incorporating the routing and traffic control into the problem scope. Scheduling and power control is also inherent to distributed computing of "global problems", *e.g.*, the maximum independent set problems in terms of transmitting links and local broadcasts respectively, and the minimum spanning tree problems. Under the physical interference model, we provide the *first* sub-linear time distributed solutions to the maximum independent set problems, and also solve the minimum spanning tree problems efficiently. We develop new techniques and algorithms and exploit the availability of technologies (full-/half-duplex radios, fixed/software-defined power control) to further improve our algorithms.

We highlight our main technical contributions, which might be of independent interest to the design and analysis of optimization algorithms. Our techniques involve the use of linear and mixed integer programs in delay-constrained throughput maximization. This demonstrates the combined use of different kinds of combinatorial optimization approaches for multi-criteria optimization. We have developed techniques for queueing analysis under general stochastic traffic to analyze network throughput and delay properties. We use randomized algorithms with rigorously analyzed performance guarantees to overcome the distributed nature of wireless data/control communications. We factor in the availability of emerging radio technologies for performance improvements of our algorithms. Some of our algorithmic techniques that would be of broader use in algorithms for the physical interference model include: formal development of the distributed computing model in the SINR model, and reductions between models of different technological capabilities, the redefinition of interference sets in the setting of SINR constraints, and our techniques for distributed computation of rulings (informally, nodes or links which are well-separated covers).

Acknowledgments

I owe much to my advisor, Dr. Anil Vullikanti, for guiding me through the academic and nonacademic endeavors. All his guidance, encouragement, patience and insights have been instrumental in shaping me through my doctoral program. His amiable nature, contagious enthusiasm, pursuit of excellence and perfection, and meticulous and sincere manner of pursuing scientific research has shaped my attitudes, set me on the path and led me to the maturity in the Ph.D. research.

I am grateful to Dr. Madhav Marathe, who has provided constant insightful advice, mentorship and help during my pursuit of the degree.

Dr. Aravind Srinivasan has been actively involved with my work. His inputs at various stages of my work have played a vital role.

I would like to thank Dr. Thomas Hou and Dr. Binoy Ravindran for giving me valuable advice and for taking the time to serve on my committee and review my work. Their work on wireless networking has always inspired me.

I spent two excellent summers respectively at the DoCoMo Labs USA in 2008 and Bell Labs in 2010, and am grateful for having worked with the professional researchers there, especially Dr. Cedric Westphal and Dr. Milind Buddhikot as my mentors.

I would like to express my gratitude to my parents and my wife Yizhou, for their constant and faithful love, unshakable support and encouragement.

Last but not least, many thanks go to and my friends and everyone at NDSSL, for their help in the process of my work and projects.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Wireless networks have witnessed a tremendous development during the past decades, and have become a ubiquitous part of modern data communication systems. At the same time, more pressing data communication demands with requirements of better throughput and lower delay from emerging applications — such as the video-streaming and traffic from the cloud in a wireless environment (*e.g.*, a cellular communication network, a wireless local access network (WLAN), a cyber-physical system (CPS)) — have over-stressed the wireless service provisioning capabilities. A key aspect of wireless communication is *interference*, which informally means that two "nearby" sender-receiver pairs cannot successfully transmit packets at the same time when they use the same frequency band. The scarcity of wireless spectrum, the strict regulation of its use, and the technical and economic limitations of the wide application of advanced hardware, have made provisioning wireless communication with large data rate and low communication delay challenging.

Understanding and improving network throughput and delay has been an active area of research in wireless networking; network control methods for scheduling, resource allocation, routing (in a multi-hop setting), and traffic control are key to the performance (*i.e.*, throughput, delay, etc.) of wireless networks. Despite advances and changes in wireless technology, the research community is still faced with many fundamental problems of resource allocation and scheduling which have yet to be fully understood. Throughput and delay objectives are generally hard (NP-hard) to optimize individually. Additionally, they are often opposing objectives: optimization for one objective can lead to unsatisfactory performance *w.r.t.* other objectives. For example, while the max-weight scheduling algorithm [132] results in optimum throughput region by max-weight scheduling (which is NP-hard), the best delay bounds known so far for max-weight scheduling depend linearly on the network size. Thus, there are tradeoffs between delay and throughput, and quantifying this tradeoff is a multi-criteria optimization problem.

Wireless networks are inherently distributed, and are becoming more so as hardware capable of initiating and forming ad hoc connections is becoming commonplace in personal computing devices, office machines, roadside devices, and sensors. Developing methods for decentralized control is an important dimension of the research in wireless networks. With the advent of wide planning and deployment of ubiquitous network connection, such as ambient intelligence (AmI) [1], the demand for practical and efficient distributed solutions becomes more pressing. This is because the network organization and even global computation sometimes need to be fulfilled without infrastructure support. A lot of the prior research on throughput-delay optimization and computation in wireless networks has been focused on centralized solutions, while developing decentralized ones remains a significant challenge.

One basic issue that fundamentally sets the design of algorithms for wireless networks apart from that for wireline networks, is *wireless interference*, which, intuitively speaking, means that two nearby links cannot transmit successfully at the same time. More accurately, wireless interference depends on the transmission decisions, the transmission power choices, and the spatial separation between transmitter-receiver pairs (and other things like environment-caused fading, which we do not consider in the dissertation). Given an arbitrary wireless network, two fundamental algorithmic aspects are *link scheduling* and *power allocation* (*i.e.*, given a subset of links, how to schedule them and allocate power levels so that the transmissions are successful); and designing efficient centralized algorithms to make optimal scheduling and power control decisions can be quite challenging. The complexity depends crucially on how interference is modeled. Most of the research in wireless networks assumes *graph-based interference* [20, 103], in which links that are within a certain distance are assumed to have conflicts, *e.g.*, the unit disk graph model, the k-hop interference model, and the protocol model. A broad class of such models is usually termed "general interference model" or "abstract interference model," where the interference relationship between links is *symmetric* and *binary*, and does not necessarily depend on geometry.

Modeling interference in a binary and localized manner, as the graph-based interference models do, greatly simplify network optimization and protocol design. However, they ignore the aggregate interference, the power control, and the applicability of the models in more realistic settings, and are known to be overly simplistic [25,46,101]. The *Physical interference model* based on "signal to interference plus noise ratio (SINR)" constraints (henceforth, referred to as the SINR model), is considered more realistic but raises more challenges for algorithm design due to the its non-local and non-linear properties. Due to the significant difference between the physical (or SINR) and the graph-based interference models, generally it is not known how to directly apply the algorithms designed for graph-based interference in an SINR setting. For example, [85] shows that the longest-queue-first scheme may result in zero throughput under SINR constraints (unlike that in the graph based model).

In this dissertation we study both graph-based and physical interference models because some of the problems have not been fully solved in the past under graph-based models, while the ultimate goal is to develop solutions for the physical interference model, which is considered more realistic.

The problems in the dissertation involve *dynamic* network traffic, which is modeled by *general* stochastic processes, and our performance metrics include *throughput* and packet delivery *delay*. As for the *throughput* performance, we do not only consider total throughput, but also the *throughput region* or *stability region* first introduced by Tassiulas and Ephremides in their seminal work [132]. The throughput region of a specific scheduling and power control algorithm is the convex hull of all arrival rate vectors, such that all the queues in the network can be stable under the algorithm. The optimum throughput region is termed *capacity region*. Achieving capacity region in many interference models involves solving an NP-hard problem at every time instance with global information [93].

Since we mainly focus on distributed algorithms, the *distributed complexity* is yet another important performance metric. We capture the distributed complexity by the number of rounds required to make a schedule for data transmission or to complete certain computation in a distributed manner. This quantifies the overhead of the time and resource spent on coordination in a decentralized environment.

The details of the above-mentioned technical background, including the performance metrics, our decision space and wireless interference, can be found in Section 2.1.

1.2 Overview of Research Problems and Results

The dissertation revolves around two sets of fundamental problems: (1) throughput and delay for dynamic traffic and (2) designing efficient distributed solutions.

In the setting of throughput-delay optimization, given a wireless network that is modeled as a graph $G = (V, L)$, and a set of sessions (or source-destination pairs), two fundamental problems are:

- to maximize the network throughput capacity, and
- to minimize the delay of communication.

The decision space includes traffic control, scheduling, routing, power control, and distributed complexity or overhead. These problems are known to be hard in general under a graph-based interference model. When we project these problems onto the physical interference model, usually they become more complicated due to the signal fading and cumulative interference throughout the network, and the non-concave form of SINR. The specific problems and our results are:

(1) the first low-complexity distributed scheduling and power control algorithms for single-hop traffic and multi-hop dynamic traffic under both graph-based interference models and the physical interference model (presented in [111, 112, 117]); and

(2) the first multi-commodity frameworks for delay-constrained throughput maximization with approximation ratios independent of the network size for the settings of unit disk graph model and cognitive networks under the physical interference model, in which we consider traffic control, routing, scheduling and power control, based on the end-to-end per-flow delay upper-bounds for our scheduling algorithms (presented in [112, 117]).

Our focus is on provable algorithms with rigorous performance guarantees. For instance, the performance results of the distributed scheduling algorithm are based on queueing analysis at packet level and capture control overhead and complete dynamics of general stochastic arrivals.

In the setting of distributed network computing for a maximum independent set and a minimum spanning tree in wireless networks, two fundamental problems are to minimize the approximation ratio and to minimize the computation time under the physical interference model. These problems are challenging in the SINR model even in a centralized setting. For example, the maximum independent set problem is NP-hard. We provide the first fast distributed algorithms for the following problems under the physical interference model:

(1) to compute a maximum independent set (MIS) of transmitting links (presented in [113, 115, 116]);

(2) to compute a maximum independent set of local broadcasting nodes (henceforth referred to as the one-shot local broadcasting problem) (presented in [116]); and

(3) to construct a minimum spanning tree (MST) (presented in [74, 75]).

Our distributed algorithms compute order-optimal solutions to the maximum independent link set and the one-shot local broadcasting problems. These serve as the first sub-linear time solutions.

Figure 1.1 provides the overview of the dissertation.

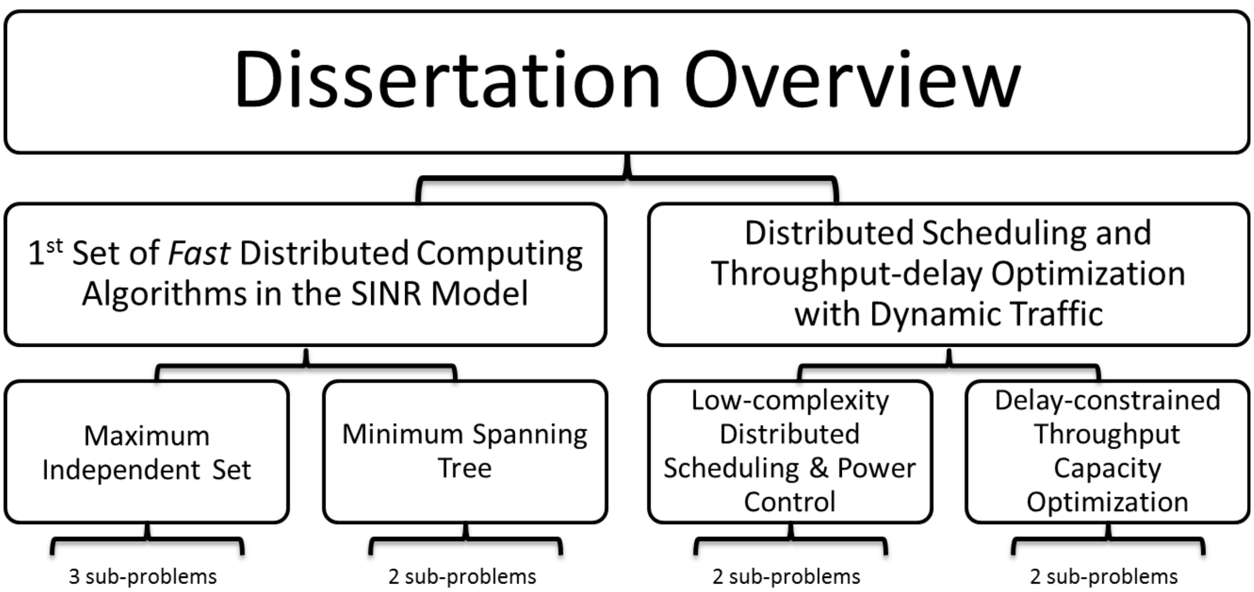


Figure 1.1: Overview.

1.3 Brief Description of Problems and Contributions

1.3.1 Scheduling and Power Control for Dynamic Traffic under the Physical Interference Model

A more detailed description of the problem and contributions can be found in Section 2.2.1. Given an arbitrary wireless network, our goal is to maximize the efficiency ratio of *scheduling* and *power assignment* for a set of flow paths carrying end-to-end dynamic traffic. That is, we design a scheduling and power assignment scheme to minimize a factor $C > 1$, such that the network is stable under an input traffic vector $\frac{\lambda}{C}$, for any vector $\lambda \in \Lambda^{OPT}$. Λ^{OPT} is the capacity region, while the factor C is the approximation factor, and $1/C$ is the efficiency ratio. Finding optimum solutions to the scheduling problem requires solving an NP-hard problem at every time slot with global information [93], *e.g.*, the back-pressure algorithm [132].

Research status quo. Many centralized scheduling schemes have been developed for graph-based interference models, and some are distributed (*e.g.*, [23,49,65,92]), but suffer from the issues of incompleteness regarding overhead analysis and unrealistic assumptions discussed in Section 1.1 of this chapter and Section 3.1.2 of Chapter 3. Under the physical interference model, traditional distributed computing models (such as the CONGEST or RBN models [118]) cannot be used. Distributed algorithms need to fully address the aggregate interference in a stochastic manner, which makes their design and analysis complex. In particular, developing efficient distributed algorithms under a formal physical interference model had been unresolved until our results.

Contributions. Knowledge of the exogenous traffic rates is pivotal in designing distributed scheduling algorithms. While this knowledge can facilitate the design of a more simple and effective scheduling algorithm, we show in Section 6.4.1 that to obtain this knowledge with high accuracy may require exponential time in the network size for some cases. We design the first low-complexity distributed scheduling and power control algorithms for dynamic traffic under the physical interference model: rate-oblivious RA-Sched-SINR for single-hop traffic (presented in Chapter 5) and rate-based RB-Sched-SINR for multi-hop traffic (presented in Section 7.3). Both RA-Sched-SINR and RB-Sched-SINR ensures a throughput region of $\Omega(\frac{1}{g(L)})\Lambda^{OPT}$, where $g(L)$ is the "link diversity" [46], which is the number of classes into which the links can be partitioned, so that links within each class have similar lengths. They use random-access based on local queue information. As an intermediate step, we solve the low-complexity distributed scheduling problems in a graph-based interference model, by designing RA-Sched and RB-Sched respectively, providing a throughput region close to $\frac{\Lambda^{OPT}}{e\mathcal{K}}$. $\mathcal{K}$ denotes the interference degree, the largest maximum number of links that can make successful transmissions simultaneously in an interference set among all of the interference sets; in many graph-based interference models (*e.g.*, unit disk graph model, primary interference model, 2-hop interference model), $\mathcal{K}$ remains a constant. Our analysis rigorously takes into account the communication complexity of each step, which is missing in the previous results. Novel technical aspects of our design include: (1) the use of infrequently-updated out-of-date information in RA-Sched-SINR to improve the efficiency, and (2) our notion of "interference sets" in the SINR setting which greatly aids the performance analysis.

1.3.2 Delay-Constrained Throughput Capacity of Wireless Networks

A more detailed description of the problems and contributions can be found in Section 2.2.2. The end-to-end delay is an important issue in many multi-hop wireless network applications, such as video streaming [79]. Additionally, there is a tradeoff between the total achievable throughput and the delays. We study the problem of computing explicit throughput-delay tradeoffs in arbitrary networks. Given a multi-hop wireless network represented by a graph $G = (\mathcal{V}, \mathcal{L})$ and a set of sessions with a target delay $\Delta(c)$ for each session c, the goal of the *Delay-constrained Throughput Maximization* (DCTM) problem is to find a stable rate vector $\lambda()$ that (approximately) maximizes the total achievable rate $\sum_c \lambda(c)$, while ensuring that the per-packet delay for session c is at most $\Delta(c)$ as possible. This problem is NP-hard, even without considering any delay guarantees [11, 135]. However, with delay constraints, this problem becomes hard to solve even approximately, which we establish later.

Research status quo. Following the seminal work of Gupta and Kumar [51], the capacity in wireless networks is fairly well understood [26,39,48,77,81,82]. However, the questions for delay-constrained throughput capacity remain unanswered, because delay analysis and optimization is already hard in itself. Delay analysis is key to determining the delay-constrained throughput capacity. We compare our scheduling and delay results with recent relevant results in Table 3.1. We note that, our random-access scheduling algorithm provides the first per-flow delay bounds independent of the network size, based on which our *unique* DCTM framework maximizes throughput with low per-session delay guarantees that only depend on target delays.

Contributions: DCTM under Graph-based Interference Models. In Chapter 6, we present the delay analysis for RB-SCHED and the delay-constrained throughput-capacity maximization framework. Our contributions include the following:

(1) *Approximation hardness of* DCTM. We show lower bounds on the computational complexity of the DCTM problem: there is a constant K such that it is NP-hard to approximate DCTM within a factor of K when the wireless network is modeled as a unit disk graph; for any constant $\varepsilon \in (0, 1)$, it is NP-hard to approximate DCTM within a factor of $O(n^{1-\varepsilon})$ in the general interference model while satisfying all delay constraints.

(2) *Multi-commodity framework for* DCTM. Given a network G, set C of sessions, target delay $\Delta(c)$ for each session c, we develop a multi-commodity flow framework to compute a rate vector $\lambda()$, routes and a synchronous random-access scheduling scheme (with an extension to asynchronous random-access) such that under the unit disk graph model: (i) total throughput capacity $\sum_c \lambda(c)$ is within a factor of $O\left(\frac{\log\log\Delta_m}{\log\Delta_m}\right)$ of the maximum possible (with the given delay constraints), where $\Delta_m = \max_c\{\Delta(c)\}$. (ii) the average end-to-end packet delay for each session c is bounded by $O\left(\left(\frac{\log\Delta_m}{\log\log\Delta_m}\Delta(c)\right)^2\right)$ (summarized in Theorems 6.4 and 6.8). These end-to-end delay guarantees include queuing delays at all intermediate nodes. The measured delay values are likely to be smaller in most practical situations.

(3) *Novel technical aspects.* Under the general interference model, when routes and mean traffic rates are fixed, we develop a sequence of *queueing system reductions* to help derive the delay bounds based on the properties of random-access scheduling with an isolation technique which brings down the dimension of the problem. Our scheduling scheme provides per-flow delay upper-bounds that depend only on a flow's path lengths, while achieving a throughput region of $\frac{1}{eI_{max}}$. In contrast, previous results (*e.g.*, [59,62,67,86]) depend on the network size or interference degrees. We use a *rounding-based algorithm* with a combination of techniques to construct a flow vector based on a novel application of the Lovász Local Lemma [131], ensuring that the factor of loss in throughput is only $O\left(\frac{\log\Delta_m}{\log\log\Delta_m}\right)$; in contrast, a straight-forward and direct application of randomized rounding [104] can only lead to an $O(\log n)$ factor.

(4) *Quantifying the impact of adaptive channel switching.* We show how to estimate throughput capacity in networks with adaptive channels and end-to-end delay requirements. These constraints can be explicitly incorporated into our framework. This differs from

the past work in that we consider both constraints of the delay and the number of channels.

(5) *Simulation results*. We study the empirical performance and properties of our algorithm and compute explicit throughput-delay tradeoffs and the saturation throughput.

DCTM under the Physical Interference Model in Cognitive Networks

Cognitive networking has been accepted and adopted as the next generation wireless communication technology due to its great potential to improve spectrum efficiency [140]. In a cognitive radio network (CRN), there are two types of users: primary and secondary users. The primary users (PU) are licensed users who have the primary right to access the wireless channels of the underlying CRN, while the secondary users (SU) are unlicensed or limited users who are allowed to use the channels (used by primary users). Our goal is to explore the opportunistic use of wireless channels for a set of secondary users under the presence of a set of primary users. The PUs already exist in the network, and there are a set $\mathcal{C}^p$ of on-going sessions with fixed routes for the PUs, and we call them "primary sessions." Each session $c \in \mathcal{C}^p$ is associated with traffic of mean rate $\lambda(c)$. The SUs have a set $\mathcal{C}^s$ of session requests, and we call them "secondary sessions." In light of these, we have the secondary-user DCTM problem (SU-DCTM) which incorporates DCTM as a special instance: Given the delay constraint $\Delta(c)$ for each secondary session request $c \in \mathcal{C}^s$, we find a stable rate vector $\lambda()$ and constructing a set of flows for each session c, such that under a distributed scheduling and power control scheme, the total throughput $\sum_c \lambda(c)$ is maximized, and the average delay requirement of $\Delta(c)$ for each session c is satisfied. In this problem, we consider the SINR-based physical interference. SU-DCTM is NP-hard and we are the first to provide a throughput-delay bi-criteria optimization framework that includes a distributed scheduling and power control scheme while optimizing routing and traffic control.

Contributions. In Chapter 7, we develop the following multi-commodity framework for SU-DCTM under the physical interference model in cognitive wireless networks using RB-SCHED-SINR as the underlying scheduling and power control scheme. Given a network G, set $\mathcal{C}^p$ of primary sessions, set $\mathcal{C}^s$ of secondary session requests, and target delay $\Delta(c)$ for each secondary session c, we develop a multi-commodity flow framework to compute a rate vector $\lambda()$, routes and a random-access scheduling scheme such that (1) the throughput rates for all the existing primary sessions are not affected; (2) total throughput capacity $\sum_{c\in\mathcal{C}^s} \lambda(c)$ is within a factor of $\Omega\left(\frac{\log\log\Delta_m}{g(L)\log\Delta_m}\right)$ of the maximum possible (with the given delay constraints), where $\Delta_m = \max_{c\in\mathcal{C}^s}\{\Delta(c)\}$, and (3) the average end-to-end packet delay for each secondary session $c \in \mathcal{C}^s$ is bounded by $O\left((\frac{g(L)\log\Delta_m}{\log\log\Delta_m}\Delta(c))^2\right)$ (summarized in Theorems 7.7 and 7.13). These end-to-end delay bounds include queuing delays at all intermediate nodes. Our results provide per-session delay bounds that depend only on the target delays and path lengths; these are likely to be smaller in most practical situations. We obtain the delay results by proving that under RB-Sched-SINR, the average delay for each flow f is $O\left(|L(f)|^2/\lambda^2(f)\right)$ when the network is stable, where $L(f)$ is the path length and $\lambda(f)$ is the arrival rate of flow f. To our knowledge, among the few results known under the physical interference model, these serve as the first set of end-to-end per-flow delay bounds related to only the path length and traffic rate of a flow for general stochastic arrival processes, and independent of the network size.

1.3.3 Distributed Computing: Maximum Independent Link Set and One-shot Local Broadcasting under the Physical Interference Model

A more detailed description of the problem and contributions can be found in Section 2.2.3. One of the most basic problems in wireless networks is to find the maximum number of active connections at the same time, that is the *Maximum Link Scheduling* problem (MaxLSP): given a set L of links, compute the largest possible subset $L' \subseteq L$ of links that can be scheduled simultaneously without conflicts; this is also referred to as the one-shot scheduling [46] or max independent link set problem [137]. Different interference models have been developed for capturing conflicts in wireless networks, most of which are based on the idea of "conflict graphs" [122]. MaxLSP is challenging under most of these models — the decision version of this problem is NP-complete under many models, and constant factor approximation algorithms are known for many interference models [122]. One related problem to MaxLSP is the *One-shot Local Broadcasting* problem (MaxLBP), in which the goal is to select a subset of nodes V' from a given set V, such that the SINR constraints are satisfied for all possible receivers within a given range of each node in V' when all nodes in V' broadcast — this generalizes MaxLSP, and is an "independent set" version of the local broadcasting scheduling problem in [45] whose goal is to make successful local broadcasts in the shortest time.

Research status quo. Since link scheduling is a common subroutine in many other problems, distributed algorithms with low complexity are crucial. Efficient (*i.e.*, polylogarithmic time) distributed algorithms have been developed for many graph based interference models. However, schedules in the graph based interference model are not feasible with respect to SINR constraints*, and a different model of distributed computing is needed for handling SINR constraints. MaxLSP and MaxLBP are algorithmically very challenging problems, and even centralized solutions to these problems are much harder in the SINR model, than in the graph based interference model. To the best of our knowledge., in the SINR model, previous distributed algorithms for MaxLSP all require at least $\Omega(n)$ time, and there is no distributed solutions to MaxLBP.

Contributions: Fast Distributed Algorithms for Constructing Uniform-power Maximum Independent Link Set under the Physical Interference Model. In Chapter 9 we develop a set of fast distributed $O(1)$-approximation for the uniform-power version of MaxLSP, in which we seek to find a maximum independent link set where all links use uniform power levels for data transmission (we refer to this as MaxLSP-U), improving the running time upon the results implied by [12, 32]. We find that different aspects of available technology, such as full/half-duplex communication, and non-adaptive/adaptive power control, have a significant impact on the performance of the algorithm; these issues have not been explored in the context of distributed algorithms in the SINR model before. Our algorithms' running time is $O(g(L)\log^c m)$, where $c = 1, 2, 3$ for different problem instances, and $g(L)$ is the "link diversity" determined by the logarithmic scale of a communication link length. Since $g(L)$ is small and remains in a constant range in most cases, our algorithms serve as the *first* set of "sublinear" time distributed solution. The algorithms are randomized and crucially use physical carrier sensing without exchanging any control information among links. One of our key technical contributions is the notion of a "ruling" (informally a spatially-separated node cover), and its distributed computation in the SINR model.

Contributions: Fast Distributed Algorithms for Constructing Uniform-power One-shot Local Broadcasting under the Physical Interference Model. In Chapter 9, we provide the *first* distributed solution to the uniform-power version of MaxLBP, which we refer to as MaxLBP-U, and demonstrate that the techniques developed for solving MaxLSP can be effective in solving other problems that were open such as MaxLBP under

*It is easy to construct instances where with $\Omega(n)$ gap between the minimum schedule lengths in the graph based and SINR models [24].

the physical interference model. The goal of MaxLBP-U is to choose a subset of nodes V' from a given set V, such that the SINR constraints are satisfied for all possible receivers within a given range of each node in V' when all nodes in V' broadcast with a uniform transmission power. Note that the targeted broadcast range can be different for different nodes although uniform power is used for broadcasting. We design a distributed constant factor approximation algorithm with running time $O(g(V)\log^2 m)$ with high probability, in a duplex model of communication, here $g(V)$ denotes the "range diversity", defined analogously to the link diversity. For a half duplex model of communication, the running time of our algorithm is $O(g(V)\log^3 m)$, w.h.p.

Contributions: Fast Distributed Algorithms for Constructing Non-uniform-power Maximum Independent Link Set under the Physical Interference Model. In Chapter 10, we develop the *first* distributed constant factor approximation algorithms for MaxLSP under non-uniform power assignments, with provable bounds on the running time and performance. We consider non-uniform length monotone sub-linear power levels (defined in Chapter 8). Our algorithm with full duplex communication finishes in time $O(g(L)\rho\log m)$, where $g(L)$ denotes the "length diversity" which refers to the number of different classes of links possible, and ρ denotes the "local density", a parameter depending on the topology — informally, ρ is the maximum number of links in L of length in the range $[d/2, d]$ such that their sender nodes fall in the same disk of any radius d. Under half duplex communication, our algorithm takes time $O(g(L)\rho\log^2 m)$. Our simulation shows that the approximation factor of our distributed algorithms for MaxLSP is within a factor of 2 of the centralized algorithm in [53].

1.3.4 Distributed Computing: Minimum Spanning Tree under the Physical Interference Model

A more detailed description of the problem and contributions can be found in Section 2.2.4. One of the most fundamental distributed computing problems in wireless networks is the *Minimum Spanning Tree (*MST*)* problem, where the cost is Euclidean distance. It is a recurring sub-problem in many network and protocol design problems, and there has been a lot of work on distributed algorithms for computing the MST. Two important applications of an MST in wireless networks are broadcasting and data aggregation to minimize the use of resources. MST is a "global" where the network diameter is an inherent lower bound.

Research status quo. We note that the known algorithms in the SINR model for spanning tree problem and the connectivity problem [15, 101, 103] are all in the centralized setting.

Contributions: Fast Distributed Algorithms for Constructing Minimum Spanning Tree under the Physical Interference Model. In Chapter 11, we we develop the *first* fast distributed approximation algorithm for MST construction in an SINR based distributed computing model. For an n-node network, our algorithm's running time is $O(D \log n + \mu \log n)$ and produces a spanning tree whose cost is within $O(\mu)$ times the optimal (MST cost), where D denotes the diameter of the disk graph obtained by using the maximum possible transmission range, and $\mu = \log \frac{d_{max}}{d_{min}}$ denotes the "distance diversity" w.r.t. the largest and smallest distances between two nodes. (When $\frac{d_{max}}{d_{min}}$ is n-polynomial, $\mu = O(\log n)$.) Our algorithm's running time is essentially optimal (up to a logarithmic factor), since computing *any* spanning tree takes $\Omega(D)$ time; thus our algorithm produces a low cost spanning tree in time only a logarithmic factor more than the time to compute a spanning tree. The distributed scheduling complexity of the spanning tree resulted from our algorithm is $O(\mu \log n)$. Our algorithmic design techniques can be useful in designing efficient distributed algorithms for related "global" problems in wireless networks in the SINR model.

1.4 Brief Highlights of Technical Contributions

Besides the results presented above, we list brief technical highlights in our contributions below, many of which may be of independent interests. Details can be found in Section 2.3.

(1) *Queueing analysis in the setting of multihop networks and SINR constraints*: we develop a sequence of reductions and combine the analyses of queueing and probabilistic SINR interference to derive throughput and delay bounds.

(2) *Concept of interference sets under the physical interference model*: we redefine the interference sets and adapt them to the context of SINR constraints, in order to facilitate the design and analysis of algorithms under the physical interference model.

(3) *Infrequently updated delayed network state information in distributed scheduling*: our approach based on infrequently updated delayed information is new and differs from previous methods of using delayed information in that it reduces the overhead of control information exchange to a negligible level and achieves a provable throughput region.

(4) *Distributed computing model*: as the existing distributed computing models (*e.g.*, the RBN model) do not capture several crucial features of SINR-based wireless networks, requiring us to rethink the design of distributed algorithms in the SINR model in a fundamentally new way, we formally define the elements and list the assumptions for distributed computing in the SINR model.

(5) *Sensing-based message-less distributed computing*: we make crucial use of physical carrier sensing in solving MaxLSP-U and MaxLBP based on the Received Signal Strength Indication (RSSI) measurement without the need of exchanging any messages.

(6) *Distributed computing sub-routine: (ω_1, ω_2)-ruling*: an (ω_1, ω_2)-ruling is informally a spatially separated node cover, and is useful for many network topology control and connection control problems; we have developed efficient distributed ruling construction algorithms under the physical interference model accounting for different availability of technologies in terms of full-/half-duplex and adaptive/non-adaptive power radios.

1.5 Dissertation Outline

In the next chapter, we detail the description of the problems and corresponding contributions, preceded by technical background; in addition, we summarize the technical highlights in our contributions. In Chapter 3 we provide the literature review for the problems mentioned above. Then, Chapter 4 and 8 present the network models, interference models, traffic models, and queueing models used in the sections following them respectively.

The research performed in this dissertation investigates optimal scheduling, power allocation, distributed computing and cross-layer throughput-delay optimization problems under the physical interference model. In Chapter 5, we present a rate-oblivious low-complexity distributed scheduling and power control algorithm RA-Sched-SINR for wireless networks under the physical interference model; as an intermediate step we present algorithm RA-Sched for the k-hop interference model. In Chapter 6, we deal with graph-based interference model and describe a rate-based distributed scheduling algorithm RB-Sched with analysis of per-flow end-to-end delay upper-bounds; based on that, we present the cross-layer optimization framework for delay-constrained throughput maximization under the unit disk graph interference model. Chapter 7 extends the optimization framework to the setting of cognitive radio networks under the physical interference model, based on the rate-based distributed scheduling and power control algorithm RB-Sched-SINR that works for multihop wireless networks under the physical interference model. Then we present the results for the distributed computing problems including MIS, MST and one-shot local broadcasting. In Chapter 9, we present fast distributed algorithms for constructing a uniform-power maximum independent link set and one-shot local broadcasting set under the physical interference model, and we explore the performance tradeoffs by designing efficient algorithms according to difference levels of technology availability. In Chapter 10, we extend our scope to the non-uniform-power version of the maximum independent link set problem and present fast distributed algorithms under the physical interference model. In Chapter 11, we present fast distributed approximation algorithms for constructing a minimum spanning tree under the physical interference model. We conclude the dissertation in Chapter 12 with final remarks.

Chapter 2

Problems and Contributions

In this chapter, we elaborate on the problem description and contributions from the previous chapter, and provide more technical details and explanations.

2.1 Technical Background

2.1.1 Wireless Interference

Interference is a basic issue in wireless networks, and the problems of *link scheduling* and *power allocation* (*i.e.*, given a subset of links, how to schedule them and allocate power levels so that the transmissions are successful) are challenging. The complexity depends crucially on how interference is modeled. Most of the research in wireless networks assumes *graph-based interference* [20, 103], in which links which are within a given distance are assumed to have conflicts. These models involve constraints that associate each link with an interference set such that the link cannot be scheduled if any other link in its interference set is scheduled. These are usually based on geometry and specify node transmission ranges and interference ranges (usually longer than transmission ranges), such as in the unit disk graph model, the k-hop interference model and protocol model; and the *interference relationship* is *symmetric* and *binary* [49, 66, 106]. We refer to a broad class of such models as "general interference model" or "abstract interference model." This model uses abstract interference sets rather than considering geometric interference constraints, such that technically any link may appear in an interference set of another. While it captures the interference in a graph-based model in a generic sense, it does not help algorithm design leverage the information of transmission ranges in graph-based interference models inferred by transmission powers.

In these graph-based models, scheduling a set of links is reduced to variants of node/edge independent sets or coloring (e.g, *strongly-induced matching* [20]); these problems are very hard in general graphs, but in geometric graphs, can be approximated within a constant factor by greedy algorithms.

By modeling interference in a binary and localized manner, the graph-based interference models greatly simplify network optimization and protocol design; however, they ignore the aggregate interference, the power control, and the applicability of the models in more realistic settings and are known to be overly simplistic [25,46,101]. Researchers in communication and information theory (the "EE community") usually use the *Physical interference model* based on "signal to interference plus noise ratio (SINR)" constraints (henceforth, referred to as the SINR model). Under this model, each link specifies its transmission power, and the strength of transmitted signals degrades with a certain factor before reaching the receiver. Signals received from all transmitters on other links and background noise together constitute the *interference*. The simultaneous transmission on a set of links is assumed to be successful, provided the ratio of the total signal strength (from the sender of each link) at each receiver to the interfering signals exceeds a given threshold. This model turns out to be analytically harder than graph based models, because of their non-locality and the added decision space of power allocation.

Due to the significant difference between the physical (or SINR) and the graph-based interference models, there exists a disappointing gap prohibiting the direct application of the graph-based algorithms in an SINR setting. For example, [85] shows that the longest-queue-first scheme may result in zero throughput under SINR constraints (unlike that in the graph based model). Most of the work in the networking community is of heuristic nature, and only in recent years provable algorithms have been developed for some problems in this model in the theoretical computer science community, such as scheduling the largest subset of links (and its variants), *e.g.*, [36, 46, 55, 69, 137], end-to-end latency minimization [25, 37], minimizing the complexity of connectivity [70,101] and finding dominating sets [125]. In this dissertation we study both graph-based and physical interference models, while the ultimate goal is to develop solutions for physical interference model.

2.1.2 Performance Metrics and Decision Space for Optimization

Network Traffic

The system operation and performance depends on the type of arrival traffic of the system. Traffic is associated with *sessions*, which can be single-hop or multi-hop. Each session corresponds to a source-destination pair where exogenous traffic enters or departs the network. Generally, arrival traffic is *dynamic*, and is modeled by stochastic processes with an average arrival rate (first moment), such that packets are continually injected at source nodes and queued at nodes before destination. Performance metrics associated with dynamic traffic usually imply quantities (in average) observed in a long term, *e.g.*, stability, throughput, average delay.

Decision Space

The network protocol stack has a layered structure, where the functions are partitioned into layers such as physical, MAC, network, transport and application. Each layer accounts for a particular type of task listed below. Due to the interdependencies across the layers, we need to employ a cross-layer approach that unifies the control across multiple layers [25,80] in order to achieve global optimal objectives. When referring to performance metrics and problem description, it is necessary to clearly state the dimension of decision space, because the corresponding optimality may change.

1. *Traffic Control* (handled by the transport layer): This is responsible for admission control or rate control at source nodes. It regulates how much arrival traffic enters the queues at source nodes, such that network does not overflow.

2. *Routing* (handled by the network layer): Each session of traffic needs to decide which path(s) to use to connect the source and destination. Each path consists of a series of links.

3. *Scheduling* (handled by the MAC layer): Wireless links are *scheduled* to access the wireless medium at each time slot. Schedulers compute a schedule (a set of active links) by either deterministic or randomized algorithms. Randomized algorithms based on random-access are widely used in a distributed setting. In the case of dynamic traffic, the knowledge of mean traffic rates is not always available: a *rate-oblivious* method refers to scheduling without using or depending on traffic rate information, *e.g.*, maximal scheduling, while a *rate-based*

method means the traffic rates are at hand for making scheduling decisions.

4. *Power Control* (handled by the physical layer): In the context of physical interference, allocating the power levels to transmitting links adds one more dimension contributing to the complexity of the scheduling problems. Since transmission on a link acts as interference ("bad") to the transmissions on other links and meaningful signal ("good") to the receiver of the link, transmission power assignment needs careful design and can impact the performance of a wireless network. We notice that a lot of existing results for physical interference assume a specific fixed power scheme and have approximation ratios that only accounts for scheduling under that specific choice of power control. The performance may degrade by a large factor with power control as an option [101]. In this dissertation, on the one hand, we develop scheduling and optimization algorithms that achieve reasonable approximation ratios by considering an optimum obtained by exploring any scheduling and power allocation schemes; on the other hand, we design distributed algorithms for widely used power assignment schemes including uniform and linear power assignments, and we exploit software-defined radios with adaptive power assignments to improve the performance of our algorithms.

5. *Multiple Radios and Channels* (handled by the physical layer): For nodes equipped with multi-radio multi-channel hardware, we are allowed (and required) to choose which frequency bands to use for wireless transmission. This is not our focus in this dissertation, although most of the results can be easily adapted to this setting.

Performance Metrics

1. *Efficiency Ratio and Throughput Capacity.*

The concept of *throughput region* or *stability region* is usually in the context of dynamic traffic which relates to scheduling, power control, radios and channels. Tassiulas and Ephremides in their seminal work [132], introduce stability region to characterize the maximum attainable network throughput and also provide a stable scheduling strategy that attains optimal throughput region in an arbitrary wireless network. Stability region or *throughput region*, denoted by Λ, is referred to as the convex hull of all arrival rate vectors that can be stably scheduled by a specific algorithm; and *capacity region*, denoted by Λ^{OPT}, refers to the maximum throughput region under any scheduling scheme. The fraction of the capacity region that a scheduling scheme achieves is called *efficiency ratio*. It is shown that achieving the capacity region in a general interference model involves solving an NP-hard problem (*e.g.*, max-weight independent set problem) at every time slot with global information [93]. The main challenges in designing stable scheduling algorithms include: (1) lowering the scheduling complexity; (2) ensuring a certain throughput region; and (3) requiring as less information as possible. The amount of information required also implies the overhead or complexity for scheduling. Under the physical interference model, scheduling must be coupled with power control which adds one important dimension to the optimization of throughput regions. Therefore, in this context, we define the throughput region based on both scheduling and power control.

Total throughput is regarded as the total mean flow rate that can be supported by a certain control policy. It corresponds to an optimizing point in the throughput region that realizes the maximum achievable total mean rate. The terms *capacity* and *capacity region* refer to optima determined by considering all operation strategies on certain layers of a network. For scheduling and power control algorithms, capacity region is the optimum throughput region that can be achieved by considering any scheduling and power control configurations; For our cross-layer optimization framework, we refer to the capacity achievable by investigating all operation options involving traffic control, scheduling, routing, power allocation. We do not address the information theoretic capacity, which includes optimization over all possible modulation and coding schemes and involves many of the unsolved problems of network information theory.

2. *Delay.*

We consider average delay for scheduling and optimization with dynamic traffic. we focus on average queueing delay which is the mean waiting time for packets to depart from the queueing system; we do not consider transmission time and node processing time. As widely known, the average queueing delay can be derived from the average queue length and the average arrival rate (which equals the average throughput rate in a stable system). We analyze the delay performance of our scheduling algorithms by bounding the average queue lengths for stable throughput rates. The traffic model we deal with is rather general and multi-hop as specified in Section 4.3.1; end-to-end delay analysis in this case is hard even in a centralized setting (as discussed in Section 3.2).

3. *Distributed Complexity.*

By *distributed complexity*, we consider the number of rounds required to make a schedule for data transmission or to complete certain computation in a distributed manner. This quantifies overhead of the time and resource spent on coordination in a decentralized environment. In this dissertation, we use distributed *overhead* and distributed complexity interchangeably. In the case of distributed computing for the maximum independent set and minimum spanning tree problems, we analyze and try to reduce the time steps needed to achieve the goal of the computation. In the case of dynamic scheduling, however, it is important that the metric of distributed complexity be coupled with throughput and delay performance metrics, because throughput rates are measured over time, *i.e.*, informally, how much can be transmitted within a certain time. The existing distributed scheduling schemes (*e.g.*, Q-SCHED [49]) usually make scheduling decisions based on information about queue sizes from interfering links. There are two main shortcomings in these results: (1) they have been developed primarily for graph based interference models, and not for the SINR model, and (2) the overhead of information exchange is ignored or not rigorously considered to be reflected in their achievable efficiency ratio. As we discuss later, the information exchange step may incur a large overhead in a distributed setting if it takes place frequently (*e.g.*, in each time slot), when the complexity is rigorously analyzed (taking into account the complete message complexity).

2.2 Description of Problems and Contributions

We provide detailed description of the problems mentioned in the previous section, each of which is followed by our corresponding results and contributions.

2.2.1 Scheduling and Power Control for Dynamic Traffic under the Physical Interference Model

Given an arbitrary multihop wireless network with a set of flow paths carrying end-to-end dynamic traffic, our goal is to maximize the efficiency ratio of *scheduling* and *power assignment*. That is, we design a scheduling and power assignment scheme to minimize a factor $C > 1$, such that for any traffic vector $\lambda \in \Lambda^{OPT}$, if it is scaled down by a factor of C, the network is stable. Λ^{OPT} is the capacity region, while the factor C is the approximation factor, and $1/C$ is the efficiency ratio. Finding optimum solutions to the scheduling problem requires solving an NP-hard problem at every time slot with global information [93], *e.g.*, the back-pressure algorithm [132]. Since [132], a lot of research efforts have been made on achieving the following three goals at the same time: (1) lowering the scheduling complexity; (2) ensuring a certain throughput region; and (3) requiring as little information as possible. The amount of information required also implies the overhead or complexity for scheduling. Based on the idea of maximal matching, a large number of algorithms are proposed, *e.g.*, variants of max-weight scheduling [93], greedy maximal scheduling [64], maximal scheduling [27, 106]. As the scheduling complexity gets improved, the achievable throughput region usually gets traded-off.

Many of the existing algorithms for link scheduling (*e.g.*, the back-pressure algorithm [132]) are centralized, and cannot or are too costly to be implemented in a practical manner in wireless networks, which are inherently distributed. Developing low complexity distributed algorithms for scheduling is a fundamental problem, and has been an active area of research. Several distributed scheduling schemes have been developed (*e.g.*, [23, 49, 65, 92]) for graph-based interference models, but suffer from the issues discussed in Section 1.1 of this chapter and Section 3.1.2 of Chapter 3.

Under the physical interference model, traditional distributed computing models (such as the CONGEST or RBN models [118]) cannot be used, since the SINR constraints need to be maintained globally at each time step. This requires more careful analysis, as done in [45, 125]. In particular, developing efficient distributed algorithms under a formal physical interference model had been unresolved until our results. Under the physical interference model, the distributed algorithms also need to fully address the aggregate interference in a stochastic manner, which makes the design and analysis involved.

Knowledge of the exogenous traffic rates is pivotal in designing distributed scheduling algorithms. While this knowledge can facilitate the design of a more simple and effective scheduling algorithm, we show in Section 6.4.1 that to obtain this knowledge with high accuracy may require exponential time in the network size for some cases. We design scheduling algorithms for both cases with and without the knowledge of traffic rates.

Contributions: Rate-oblivious Low-complexity Distributed Scheduling and Power Control for Dynamic Traffic

We present in Chapter 5 the first low-complexity distributed scheduling and power assignment algorithm (called RA-SCHED-SINR) in the SINR model which ensures a throughput region of $\Omega(\frac{1}{g(L)})\Lambda^{OPT}$, where $g(L)$ is the "link diversity" [46], which is the number of classes into which the links can be partitioned, so that links within each class have similar lengths. RA-SCHED-SINR uses random-access based on local queue information. Our analysis rigorously takes into account the communication complexity of each step. A novel aspect of our analysis is the use of out-of-date information to significantly improve the efficiency.

As an intermediate step, we first design a low-complexity distributed scheduling algorithm RA-SCHED in a graph-based interference model, which is of independent interest and helps in the analysis of the more complex RA-SCHED-SINR. Our algorithm builds on the Q-SCHED protocol of [49] and utilizes neighborhood queue size information for scheduling, but is simpler because of its explicit use of random-access. Other distinguishing features of RA-SCHED and its analysis include: (1) Links access the channel in a slotted Aloha fashion, which do not involve per-slot control phase; (2) Links make only infrequent local queue size information update, and use stale information for scheduling; rather than assuming a "magically" done per-slot queue size information exchange, we prove that the throughput loss due to the info-exchange process can be arbitrarily low. The achievable throughput region of RA-SCHED is close to $\frac{\Lambda^{OPT}}{e\mathcal{K}}$, where $\mathcal{K}$ is the interference degree defined in Chapter 4 which is the largest maximum number of links that can make successful transmissions simultaneously in an interference set among all of the interference sets; in many graph-based interference models, $\mathcal{K}$ remains a constant.

RA-SCHED-SINR is based on RA-SCHED, with a different notion of "interference sets," which leads to slightly simpler sufficiency conditions for the throughput region than [24]. Our power control policy is simple and flexible: we only require links of similar lengths use similar power values. This policy incorporates many oblivious power assignments as special cases. The scheduling is implemented with a frame structure where each frame consists of two multi-slot sub-frames for control and data transmission separately. Our results take the background noise fully into account.

Contributions: Rate-based Low-complexity Distributed Scheduling and Power Control for Dynamic Traffic

First, we describe in Section 6.4 the rate-based scheduling algorithm RB-Sched that obtains a throughput region of $\frac{1}{e\mathcal{K}}\Lambda^{OPT}$ under the abstract graph-based model, and the discuss the length of time needed to obtain accurate estimate of traffic rates.

Then, we present in Section 7.3 the first low-complexity distributed scheduling and power assignment algorithm (called RB-Sched-SINR) for multihop wireless networks with end-to-end sessions of dynamic traffic in the SINR model, achieving a throughput region of $\Omega(\frac{1}{g(L)})\Lambda^{OPT}$. RB-Sched-SINR uses random-access based on the knowledge of local traffic rate without control message passing among links. The power control policy only requires links of similar lengths use similar power values. This policy incorporates uniform, square-root and linear power assignments as special cases. The design and analysis for RB-Sched serve as an intermediate step for RB-Sched-SINR; however, accounting for SINR constraints in the throughput analysis of the statistical behavior of the queues under the scheduling policy and embedding $g(L)$ in the throughput analysis make it quite involved. The efficiency ratios of RB-Sched-SINR and RA-Sched-SINR are in the same order, but they represent different tradeoffs between applicability and control complexity. While RA-Sched-SINR does not impose such a limitation as RB-Sched-SINR on its application scope that the traffic rates should be known or should be able to be accurately estimated, RA-Sched-SINR involves more control complexity: RA-Sched-SINR uses a frame structure, and under RA-Sched-SINR the probability of channel access changes with queueing states while information exchange among neighboring links are needed in every frame. .

2.2.2 Delay-Constrained Throughput Capacity of Wireless Networks

Throughput capacity is an important fundamental problem of wireless networks that involves the following two questions: "*What* is the throughput capacity of an arbitrary network?" and "*How* this capacity can be achieved?"The throughput capacity problem mentioned in [44] does not fall in this line, rigorously speaking. In fact, it is exactly the non-weighted MIS problem. Following the seminal work of Gupta and Kumar [51], the capacity in wireless networks is fairly well understood [26, 39, 48, 77, 81, 82]. However, the questions for delay-constrained throughput capacity remain unanswered, because delay analysis and optimization is already hard in itself. We study the problem of determining and achieving delay-constrained throughput capacity, which involves an optimization space of choosing a rate vector (*i.e.*, traffic control), finding paths for each session (*i.e.*, routing), scheduling, power control at the same time.

The end-to-end delay is an important issue in many multi-hop wireless network applications, such as video streaming [79], and there is a tradeoff between the total achievable throughput and the delays; an important open question in this area has been to decide if it is possible to achieve delays proportional to the number of hops for each session, without much loss in throughput or throughput region [59]. Here we study the problem of computing explicit throughput-delay tradeoffs in arbitrary networks. Given a multi-hop wireless network represented by a graph $G = (\mathcal{V}, \mathcal{L})$ and a set of sessions with a target delay $\Delta(c)$ for each session c, the goal of the *Delay-constrained Throughput Maximization* (DCTM) problem is to find a stable rate vector $\lambda()$ that (approximately) maximizes the total achievable rate $\sum_c \lambda(c)$, while ensuring that the per-packet delay for session c is at most $\Delta(c)$ as possible. This problem is NP-hard, even without considering any delay guarantees [11,135]; however, with delay constraints, this problem becomes hard to solve even approximately, as we establish later. Therefore, we study "bi-criteria" approximation algorithms which approximately maximize the total throughput, while allowing the delay constraints to be relaxed by a certain factor; our focus is on designing algorithms with provable approximation guarantees.

While there has been a lot of work on delay-throughput tradeoffs, especially for random networks or restricted 1-hop sessions, the best bounds for end-to-end delays known so far are by [59, 62, 86]. In this dissertation, we develop an algorithm for DCTM based on our scheduling scheme that improves on these bounds.

Delay-Constrained Throughput-capacity Maximization under the Physical Interference Model in Cognitive Networks

Cognitive networking has been accepted and adopted as the next generation wireless communication technology due to its great potential to improve spectrum efficiency [140]. In a cognitive radio network (CRN), there are two types of users: primary and secondary users. The primary users (PU) are licensed users who have the primary right to access the wireless channels of the underlying CRN, while the secondary users (SU) are unlicensed or limited users who are allowed to use the channels (used by primary users) only when their existence do not affect the performance of primary users' applications. Figure 2.1 shows an example scenario of CRN.

Under this cognitive radio setting, our goal is to explore the opportunistic use of wireless channels for a set of secondary users under the presence of a set of primary users, to achieve more efficient use of spectrum and deployment of wireless systems with minimal infrastructure support. The primary users already exist in the network, and there are a set $\mathcal{C}^p$ of on-going sessions with fixed routes for the primary users, and we call them "primary sessions." Each session $c \in \mathcal{C}^p$ is associated with traffic of mean rate $\lambda(c)$. The secondary users have a set $\mathcal{C}^s$ of session requests, and we call them "secondary sessions." In light of these, we have the secondary-user DCTM problem (SU-DCTM) which incorporates DCTM as a special instance: Given the delay constraint $\Delta(c)$ for each secondary session request $c \in \mathcal{C}^s$, we find a stable rate vector $\lambda()$ and constructing a set of flows for each session c, such that under a distributed scheduling and power control scheme, the total throughput $\sum_c \lambda(c)$ is maximized, and the average delay requirement of $\Delta(c)$ for each session c is satisfied. In this problem, we consider the SINR-based physical interference.

SU-DCTM is NP-hard and this follows from the fact that even without any delay constraints and the existence of any primary users, the throughput maximization problem under the physical interference model is NP-hard [46]. Despite the large body of research work for studying the throughput capacity of CRNs on cooperative use of wireless spectrum, multi-channel multi-radio capacity, we are the first to provide a throughput-delay bi-criteria optimization framework that includes a distributed scheduling and power control scheme with optimizing routing and traffic control.

Contributions: DCTM under Graph-based Interference Models

In Chapter 6, we present the delay analysis for RB-Sched and the delay-constrained throughput-capacity maximization framework. Our detailed contributions include the following:

1. *Approximation hardness of* DCTM. We show lower bounds on the computational complexity of the DCTM problem. When the wireless network is modeled as a unit disk graph, we show that there is a constant K such that it is NP-hard to approximate the DCTM problem within a factor of K; in the general interference model, DCTM is hard to ap-

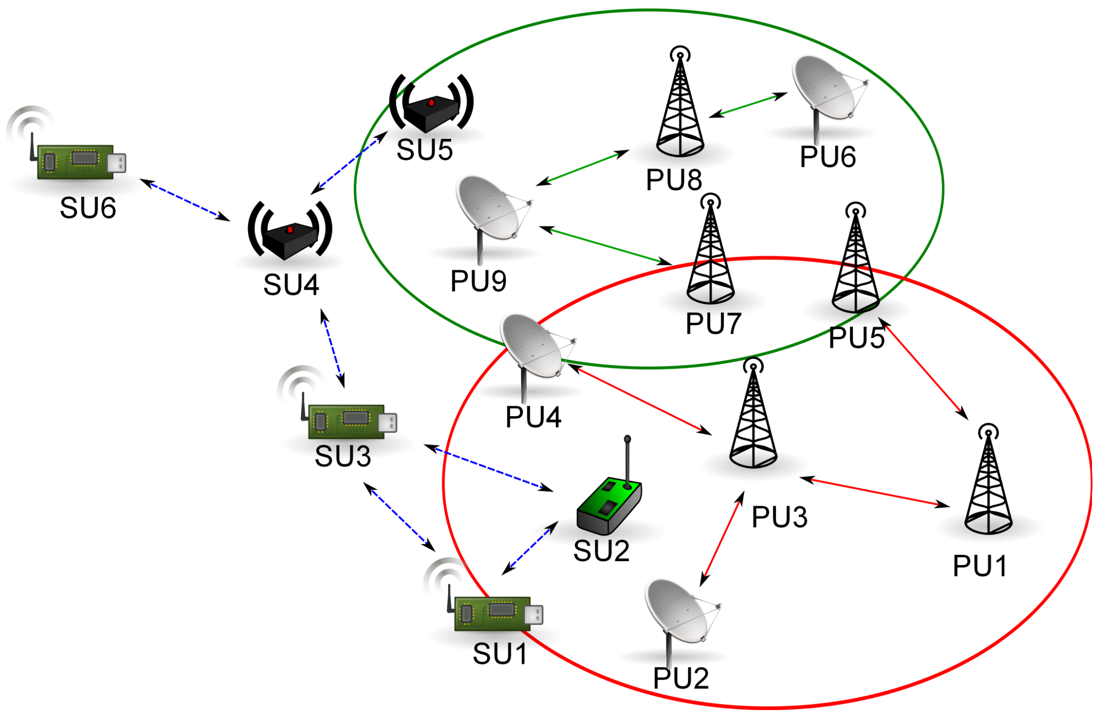


Figure 2.1: An example of a cognitive radio network. The devices labeled with PU are the primary users; the devices labeled with SU are the secondary users. The solid lines with arrows indicate existing primary-user links and the dotted lines with arrows indicate secondary-user links.

proximate within a factor of $O(n^{1-\varepsilon})$ for any constant $\varepsilon \in (0,1)$, while satisfying all delay constraints. In light of these hardness results, we need to relax the delay constraints: we study bi-criteria algorithms that simultaneously approximately maximize throughput and guarantee the delay performance.

2. *Multi-commodity framework for* DCTM. Given a network G, set C of sessions, target delay $\Delta(c)$ for each session c, we develop a multi-commodity flow framework to compute a rate vector $\lambda()$, routes and a synchronous random-access scheduling scheme (with an extension to asynchronous random-access) such that under the unit disk graph model: (1) total throughput capacity $\sum_c \lambda(c)$ is within a factor of $O\left(\frac{\log\log\Delta_m}{\log\Delta_m}\right)$ of the maximum possible (with the given delay constraints), where $\Delta_m = \max_c\{\Delta(c)\}$, and (2) the average end-to-end packet delay for each session c is bounded by $O\left((\frac{\log\Delta_m}{\log\log\Delta_m}\Delta(c))^2\right)$ (summarized in Theorems 6.4 and 6.8). These end-to-end delay guarantees include queuing delays at all intermediate nodes.

Our result involves two basic steps: (1) We use RB-Sched as the underlying scheduling algorithm. We derive end-to-end delay upper-bounds for RB-Sched under the general interference model, with a form of $O(|L(f)|^2/\lambda^2(f))$, where $|L(f)|$ is the path length and $\lambda(f)$ is the traffic rate of flow f. This is the first per-flow delay bounds independent of network size in graph-based interference models for end-to-end dynamic traffic with general arrival processes, in contrast to previous results (*e.g.*, [59,62,67,86]) that depend on the network size or interference degrees. This has motivated the design and techniques in the later rounding-based algorithm for traffic control and routing. (2) We use a rounding-based algorithm for a linear-programming (LP) relaxation of the problem to construct a flow vector that uses "short" paths (or has "low" costs), to send "high" flow on each selected path. Our algorithm is based on a novel application of the Lovász Local Lemma [131], combined with filtering and path refining steps in order to reduce certain kinds of dependencies. Our specific rounding scheme is crucial in ensuring that the factor of loss in throughput is only $O\left(\frac{\log\Delta_m}{\log\log\Delta_m}\right)$; in contrast, a straight-forward and direct application of randomized rounding [104] can only lead to an $O(\log n)$ factor. Path-constrained flows have been studied in a wired setting, *e.g.*, [19, 52]; interference constraints and the fact that we need both short paths and large flow values makes our problem different.

3. *Quantifying the impact of adaptive channel switching.* One of the novel aspects of cognitive networks is adaptive channel switching; however, this can add to the delays [84]. As a specific application of our approach, we show how to estimate throughput capacity in networks with

adaptive channels (*e.g.*, in cognitive networks) and end-to-end delay requirements. These constraints can be explicitly incorporated into our framework, thereby allowing us to provably quantify the tradeoffs between these constraints. Much of the work dealing with these aspects (*e.g.*, [39, 83]) has only considered individual constraints such as the delay or the number of channels; our approach allows all of them to be incorporated simultaneously.

4. *Simulation results*. We study the empirical performance of our algorithm on small networks and compute explicit throughput-delay tradeoffs and the saturation throughput for given delay bounds. For multi-channel networks we observe that there is significant tradeoff between the number of channels, the delays, and total throughput rate. In particular, for a given target delay, there is a threshold beyond which additional channels do not help. In our experiments, we also examine the end-to-end delays with various indices. We find our analytical delay bounds turn out to be rather conservative, *i.e.*, our scheduling schemes may perform much better in real-world instances.

Contributions: DCTM under the Physical Interference Model in Cognitive Networks

In Chapter 7, we develop the following multi-commodity framework for SU-DCTM under the physical interference model in cognitive wireless networks using RB-SCHED-SINR as the underlying scheduling and power control scheme.

Given a network G, set $\mathcal{C}^p$ of primary sessions, set $\mathcal{C}^s$ of secondary session requests, and target delay $\Delta(c)$ for each secondary session c, we develop a multi-commodity flow framework to compute a rate vector $\lambda()$, routes and a random-access scheduling scheme such that (1) the throughput rates for all the existing primary sessions are not affected; (2) total throughput capacity $\sum_{c\in\mathcal{C}^s}\lambda(c)$ is within a factor of $\Omega\left(\frac{\log\log\Delta_m}{g(L)\log\Delta_m}\right)$ of the maximum possible (with the given delay constraints), where $\Delta_m = \max_{c\in\mathcal{C}^s}\{\Delta(c)\}$, and (3) the average end-to-end packet delay for each secondary session $c \in \mathcal{C}^s$ is bounded by $O\left(\left(\frac{g(L)\log\Delta_m}{\log\log\Delta_m}\Delta(c)\right)^2\right)$ (summarized in Theorems 7.7 and 7.13). These end-to-end delay bounds include queuing delays at all intermediate nodes. Our results provide per-session delay bounds that depend only on the target delays and path lengths; these are likely to be smaller in most practical situations.

We construct our optimization framework in a similar bottom-up manner following the basic work flow in Chapter 6 and account for SINR constraints:

(1) We employ RB-Sched-SINR as our underlying scheduling and power control algorithm, such that the secondary users may access the frequency and coexist with the primary users without undermining the existing primary-user throughput. We provide rigorous analysis of the end-to-end delay upper-bounds for RB-Sched-SINR in terms of the path lengths of the flows. Our main technical contribution in the derivation of delay bounds includes a sequence of reductions to a simpler queuing system whose delay can be bounded by Lyapunov analysis — these reductions crucially uses the properties of random-access scheduling with an isolation technique which brings down the dimension of the problem and which would be of independent interest. We are able to show that the delay bounds are not impacted by the "schedule augmentation" (defined in Chapter 4) which partitions the links into $g(L)$ link classes and schedule only one link class at each time step. The average delay for each flow f is $O\left(|L(f)|^2/\lambda^2(f)\right)$ when the network is stable; the average network delay is $O\left(\sum_{f\in\mathcal{F}}|L(f)|^2\Big/\left(\sum_{f\in\mathcal{F}}\lambda(f)\min_{f\in\mathcal{F}}\{\lambda(f)\}\right)\right)$, where $L(f)$ is the path length and $\lambda(f)$ is the arrival rate of flow f. To our knowledge, among the few results known under the physical interference model, these serve as the first set of end-to-end per-flow delay bounds related to only the path length and traffic rate of a flow for general stochastic arrival processes, and independent of the network size.

(2) Based on the end-to-end delay results, we formulate and solve the optimization programs for controlling traffic rates and finding flow paths for the secondary users under the physical interference model. We reuse the techniques invented in Chapter 6 based on a novel application of the Lovász Local Lemma [131], while carefully ensuring that the SINR constraints are respected in each of the steps and accounting for the $g(L)$ factor involved in the dynamic schedule based on RB-Sched-SINR. In the end, the approximation ratio for the secondary-user throughput-capacity is $O\left(\frac{g(L)\log\Delta_m}{\log\log\Delta_m}\right)$ while the throughput rates for existing primary users remain unchanged.

2.2.3 Distributed Computing: Maximum Independent Link Set and One-shot Local Broadcasting under the Physical Interference Model

One of the most basic problems in wireless networks is to find the maximum number of active connections at the same time, that is the *Maximum Link Scheduling* problem (MaxLSP):

given a set L of links, compute the largest possible subset $L' \subseteq L$ of links that can be scheduled simultaneously without conflicts; this is also referred to as the one-shot scheduling [46] or max independent link set problem [137]. Different interference models have been developed for capturing conflicts in wireless networks, most of which are based on the idea of "conflict graphs" [122]. MaxLSP is challenging under most of these models — the decision version of this problem is NP-complete under many models, and constant factor approximation algorithms are known for many interference models [122].

One related problem to MaxLSP is the *One-shot Local Broadcasting* problem (MaxLBP), in which the goal is to select a subset of nodes V' from a given set V, such that the SINR constraints are satisfied for all possible receivers within a given range of each node in V' when all nodes in V' broadcast — this generalizes MaxLSP, and is an "independent set" version of the local broadcasting scheduling problem in [45] whose goal is to make successful local broadcasts in the shortest time.

Since link scheduling is a common subroutine in many other problems, distributed algorithms with low complexity are crucial. Efficient (*i.e.*, polylogarithmic time) distributed algorithms have been developed for many graph based interference models (generically referred to as the "Radio Broadcast Network (RBN)" model) for fundamental problems, including independent sets, coloring and dominating set, *e.g.*, [100, 126]. In the RBN model, a receiver faces a collision if transmissions from multiple senders reach it simultaneously. Algorithms in this model rely on having spatial separation between senders to ensure interference free communication; however, schedules in the RBN model are not feasible with respect to SINR constraints*, and a different model of distributed computing is needed for handling SINR constraints. This is especially important in emerging technologies such as cognitive networks, where secondary (unlicensed) users need to be scheduled with given constraints that do not disrupt primary (licensed) users [7]. Primary user constraints can be modeled by specifying the regions around them within which the SINR needs to be lower than some threshold—this is shown by the disks in Figure 2.2. The set V represents secondary users, which have to sense the spectrum and use it opportunistically, which motivates a non-uniform power assignment. In such settings, the MaxLSP and MaxLBP problems necessarily need to be solved in a distributed manner. Indeed, the gains from cognitive networks are unlikely to be realized unless these basic scheduling problems can be implemented with low complexity.

*It is easy to construct instances where with $\Omega(n)$ gap between the minimum schedule lengths in the graph based and SINR models [24].

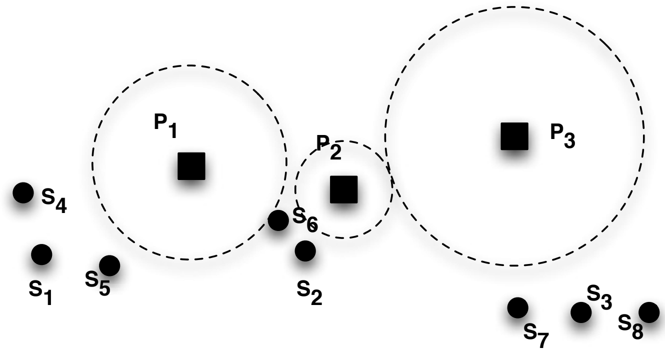


Figure 2.2: Scheduling constraints in cognitive networks: the nodes P_1, P_2, P_3 represent primary users, whereas nodes $S_1, \ldots, S_8$ represent secondary users. The disks around the primary users represent regions where the SINR constraints need to be higher than some threshold.

MaxLSP and MaxLBP are algorithmically very challenging problems, and even centralized solutions to these problems are much harder in the SINR model, than in the graph based interference model; recent work by [44, 53, 55, 69, 137] gives constant factor approximation algorithms for various instances of MaxLSP in the SINR model. The centralized algorithms of [44, 53, 55, 137] are based on a greedy ordering of the links, which requires estimating the "affectance," (which, informally, is a measure of interference) at each stage (this is discussed formally later in Chapter 8) — this is one of the challenges in distributed solutions to MaxLSP. We note that efficient time distributed algorithms for scheduling all the links (*i.e.*, the coloring version) is already known [54]. Adapting them would immediately yield a distributed $O(\log m)$-approximation to MaxLSP, but it is not clear how to obtain a distributed $O(1)$-approximation. No efficient (sub-linear) time distributed algorithms are known for MaxLSP even for the special case of uniform power levels, when each link l uses the same transmission power level. No results are known for the MaxLBP in the SINR model, to the best of our knowledge.

Contributions: Fast Distributed Algorithms for Constructing Uniform-power Maximum Independent Link Set under the Physical Interference Model

In Chapter 9 we develop a set of fast distributed $O(1)$-approximation for the uniform-power version of MaxLSP, in which we seek to find a maximum independent link set where all links use uniform power levels for data transmission (we refer to this as MaxLSP-U), improving the running time upon the results implied by [12, 32]. Our algorithms and the proofs build on ideas from [44, 55, 137] and [125], and one of the key technical contributions of our work is the notion of a "ruling", and its computation in the SINR model (discussed below). Our results raise two new issues in the context of distributed algorithms in the SINR model — adaptive power control (*i.e.*, the feature of using lower than the maximum power level, as needed), and full/half duplex communication (*i.e.*, whether nodes can transmit and receive simultaneously in the same frequency). We find these features impact the performance of the algorithms quite a bit. We summarize some of the key aspects of the results and main challenges below.

(1) In the case of "non-adaptive power control" (*i.e.*, if all nodes are required to use fixed uniform power levels), we design a distributed algorithm that provably runs in $O(g(L)\log^3 m)$ time and gives an $O(1)$ approximation to the optimum solution for half duplex communication, and we improve the running time to $O(g(L)\log^2 m)$ for the case of full duplex communication; here $g(L)$ denotes the "link diversity", which is the logarithm of the ratio of the largest to the smallest link length (this is defined formally in Chapter 8). If nodes are capable of "adaptive power control" (*i.e.*, they can use varying power levels for scheduling, but not data transmission), we improve the running time of the above algorithm to $O(g(L)\log^2 m)$ time for half duplex communication, and $O(g(L)\log m)$ time for full duplex communication. Note that in the adaptive power control case, the algorithm uses varying power levels during its run, but the links which are selected finally use the fixed uniform power level for data transmission.

(2) One of our key ideas is the parallelization of the centralized link selection step of [44, 55, 137], which requires sorting all links, processing a larger set of links simultaneously, and efficient filtering based on spatial and interference constraints in parallel. Moreover, it turns out that the usual notion of independence based on spatially separated nodes is inadequate because of the spatial separation of the sender and the receiver of a link: it is the senders which makes the distributed decision of transmission and the participation in the independent set, while the SINR model is receiver-oriented and it is hard for each sender of a candidate link to determine the interference caused by the chosen links at the corresponding receiver. One of the important steps of our algorithm involves the distributed construction of a "ruling" (first introduced in [31]) which relates to the notion of independence and aids the solution to MIS and coloring problems in graph topologies [17, 109]. The extension of the notion of ruling and its computation in the SINR model is one of the important technical

contributions. We believe this basic construct would be useful in other link and topology control problems.

(3) Our algorithms crucially use physical carrier sensing and the distributed decisions are made based on the Received Signal Strength Indication (RSSI) — this approach was used by Scheideler et al. [125] for distributed constant density dominating set construction. This aspect is another distinction from distributed algorithms in the RBN model. Further, our algorithm uses constant size messages, and all the steps can be implemented within the model without any additional capabilities or assumptions (*e.g.*, such as those made in [12]).

Contributions: Fast Distributed Algorithms for Constructing Uniform-power One-shot Local Broadcasting under the Physical Interference Model

In Chapter 9, we also investigate the *One-shot Local Broadcasting* problem (MaxLBP) and demonstrate that the techniques developed for solving MaxLSP can be effective in solving other problems that were open such as MaxLBP under the physical interference model. We solve the uniform-power version of MaxLBP, which we refer to as MaxLBP-U. The goal of MaxLBP-U is to choose a subset of nodes V' from a given set V, such that the SINR constraints are satisfied for all possible receivers within a given range of each node in V' when all nodes in V' broadcast with a uniform transmission power. Note that the targeted broadcast range can be different for different nodes although uniform power is used for broadcasting.

We design a distributed constant factor approximation algorithm with running time $O(g(V)\log^2 m)$ with high probability, in a duplex model of communication, here $g(V)$ denotes the "range diversity", defined analogously to the link diversity. Our approach involves adapting the approach of [53, 55, 114]. Further, for a half duplex model of communication, the running time of our algorithm is $O(g(V)\log^3 m)$, w.h.p. Our results for ruling construction functions effectively again as one of the core building block of the distributed solution.

Contributions: Fast Distributed Algorithms for Constructing Non-uniform-power Maximum Independent Link Set under the Physical Interference Model

In Chapter 10, we develop the first distributed constant factor approximation algorithms for MaxLSP under non-uniform power assignments, with provable bounds on the running time and performance. We consider non-uniform length monotone sub-linear power levels (defined in Chapter 8). We find that the performance bounds of our algorithms depend crucially on whether or not the devices have the capability for full/half duplex communication. We summarize some of the key aspects of the results and main challenges below.

(1) For MaxLSP with length-monotone sub-linear power levels, we design a distributed algorithm that runs in time $O(g(L)\rho \log m)$, for full duplex communication in the SINR model; here $g(L)$ denotes the "length diversity" which refers to the number of different classes of links possible, and ρ denotes the "local density", a parameter depending on the topology — informally, ρ is the maximum number of links in L of length in the range $[d/2, d]$ such that their sender nodes fall in the same disk of any radius d. In practice, and in instances such as random distributions, this parameter is at most $O(\log m)$. When the communication model is the more restrictive but more realistic half duplex model, our algorithm takes time $O(g(L)\rho \log^2 m)$.

(2) We study the performance of our algorithms through simulations. We find that the approximation factor of our distributed algorithms for MaxLSP is within a factor of 2 of the centralized algorithm in [53]. Thus, the distributed ruling step, which selects a subset of links from the same length class, does not significantly affect the quality of the solution. Further, we find that the topology has a significant impact on the performance.

Key technical contributions for MaxLSP. The key aspect of our distributed algorithms is the distributed implementation of the affectance computation step. We adapt the notion of "ruling" developed in [114] to compute the affectance bounds for all links in a given length class. Our algorithms crucially use physical carrier sensing and the distributed decisions are made based on the Received Signal Strength Indication (RSSI) — this approach was used by [114,125] for distributed algorithms for constant density dominating set construction and MaxLSP for uniform power levels. Further, our algorithms use constant size messages, and all the steps can be implemented within the model without any additional capabilities or assumptions (*e.g.*, such as those made in [12]).

2.2.4 Distributed Computing: Minimum Spanning Tree under the Physical Interference Model

Emerging networking technologies such as ad hoc wireless and sensor networks operate under inherent resource constraints such as power, bandwidth etc. A distributed algorithm which exchanges a large number of messages and takes a lot of time can consume a relatively large amount of resources, and is not very suitable in a resource-constrained network. Also, the topology of these networks can change dynamically. Communication cost and running time is especially crucial in a dynamic setting. Hence it becomes necessary to design efficient distributed algorithms for various network optimization problems that have low communication and time complexity, even possibly at the cost of a reduced quality of solution. (For example, there is not much point in having a deterministic optimal algorithm if it takes too much time, since the topology could have changed by that time.) For this reason, in the distributed context, such algorithms are motivated even for network optimization problems that are not NP-hard, *e.g.*, minimum spanning tree (MST), shortest paths (see *e.g.*, [35]). However, much of the theory of distributed approximation for various fundamental problems such as MST have been developed in the context of wired networks, and not for wireless networks under a realistic interference model.

One of the most fundamental distributed computing problems in wireless networks is the *Minimum Spanning Tree (*MST*)* problem. It is a recurring sub-problem in many network and protocol design problems, and there has been a lot of work on distributed algorithms for computing the MST. Computing an MST by a distributed algorithm is a fundamental task, as the following distributed computation can be carried over the best backbone of the communication graph. Two important applications of an MST in wireless networks are broadcasting and data aggregation. An MST can be used as broadcast tree to minimize energy consumption since it minimizes $\sum_{(u,v)\in T} d^{\alpha}(u, v)$. It is shown in [9, 30, 134] that broadcasting based on MST consumes energy within a constant factor of the optimum.

Centralized algorithms for various fundamental problems, such as independent sets, coloring, dominating set have been developed in this model in recent years, *e.g.*, [44,55,137]. However, distributed algorithms which satisfy the SINR constraints at each step are only known for very few problems (*e.g.*, [37, 54, 70]). Furthermore, the distributed algorithms known are for “local” problems such as independent set and coloring and not for “global” problems. Global problem are those that require an algorithm to “traverse” the entire network. Classical "global" problems include spanning tree, minimum spanning tree, shortest path etc. Network diameter is an inherent lower bound for such problems. We note that the known algorithms in the SINR model for spanning tree problem and the connectivity problem [15,101,103] are in the centralized setting.

Contributions: Fast Distributed Algorithms for Constructing Minimum Spanning Tree under the Physical Interference Model

In Chapter 11, we propose the first distributed algorithms under the physical interference model for approximate MST construction with cost $O(\mu)$ relative to that of the MST, where $\mu = \log \frac{d_{max}}{d_{min}}$ denotes the "distance diversity" w.r.t. the largest and smallest distances between two nodes. The running time is within a polylogarithmic time of the lower bound. We discuss our results in details below.

(1) The running time of our algorithm is $O(D \log n + \mu \log n)$, with high probability, where D is the diameter of the graph restricted by the maximum range. This is optimal up to a polylogarithmic factor, since computing *any* spanning tree takes $\Omega(D)$ time. In particular, if the maximum power level is unconstrained, the running time is $O(\mu \log n)$. The spanning tree produced by our algorithm has a low distributed scheduling complexity (defined in Section 11.4.4) of $O(\mu \log n)$, *i.e.*, the transmission requests on the edges can be scheduled distributedly in $O(\mu \log n)$ time steps, for any orientation of the edges.

(2) Our main technical contribution is the adaptation of the technique of "Nearest Neighbor Trees" (NNT) [71,72,73] to the SINR model. This technique results in spatial separation at each step, which helps in ensuring the SINR constraints. Our algorithmic design technique can be useful in designing efficient distributed algorithms for related "global" problems in wireless networks in the SINR model.

All prior distributed algorithms in the SINR model have been restricted to scheduling kind of problems (*e.g.*, independent sets, coloring and broadcasting). Our result is the first distributed algorithm for a problem not in the above class, and might be useful in other network design problems.

2.3 Highlights and Discussion of Technical Contributions

The focus of our work is theoretical, and our main results include the development of distributed techniques in the SINR model, a formal study of the impact of technological aspects of wireless capabilities on distributed algorithms, the optimization framework, etc. Now we summarize the technical highlights in our contributions.

2.3.1 Queueing Analysis in the Setting of Multihop Networks and SINR Constraints

Besides dealing with the physical interference model, we also make contributions in developing techniques for queueing analysis. Our traffic model involves general arrival processes, which is much harder to be analyzed when there are multi-hop queues and interfering traffic than Poisson or Geometric arrival processes. In Chapter 6 we develop a sequence of reductions to a simpler queuing system whose delay can be bounded by Lyapunov analysis — these reductions crucially use the properties of random-access scheduling with an isolation technique which brings down the dimension of the problem.

In Chapter 5 we combine the analyses of queueing and probabilistic SINR interference, since the probability for nodes' random medium access and the queueing states crucially depends on each other; this is the first of such work in a distributed setting, and it is challenging because the interference relationship is no longer binary and symmetric in contrast to that in graph-based interference model and adds to more complicated statistical analysis.

2.3.2 Concept of Interference Sets under the Physical Interference Model

The concept of "Interference sets" is an essential part in graph-based interference models. We redefine the interference sets and adapt them to the context of SINR constraints, in order to facilitate the design and analysis of algorithms under the physical interference model. We find that centralized algorithms, such as the maximum weighted scheduling algorithm [132] and maximum scheduling algorithm [27, 106] which were design to work in abstract graph-based interference models can be adapted to the SINR setting, using the augmented scheduling described in 4 and a simple power assignment scheme where the transmission powers of links in the same link class are within a constant factor of each other. The adapted maximal scheduling algorithms can achieve a throughput region of $\Omega(\frac{1}{g(L)})\Lambda^{OPT}$ under the physical interference model in contrast to the $\frac{1}{\mathcal{K}}\Lambda^{OPT}$ throughput region which can be potentially $\frac{1}{n}\Lambda^{OPT}$ under the abstract graph-based interference model. Note that $g(L)$ which is typically small (< 20) as discussed in Chapter 4.

In the design of distributed scheduling algorithm with SINR constraints, we make extensive use of randomized schemes to lower the complexity caused by control message passing. The concept of interference set under the physical interference model does not affect the design of our distributed algorithms; it aids in the analysis of the success rate of distributed data communication for our scheduling algorithms and the probabilistic analysis of signal strength for our distributed computing algorithms.

2.3.3 Infrequently Updated Delayed Network State Information in Distributed Scheduling

Usually, scheduling algorithms depend on network state information (*e.g.*, queueing information) immediately available in every time slot to compute efficient schedules. The network state information may include global information or neighborhood information. In a distributed setting, exchanging this information among nodes could incur high overhead, for example, even when the information is exchanged among neighbors, it could take as large as $\Omega(n)$ steps for all the nodes to successfully transmit/receive messages to/from their neighbors when the network is dense. This overhead is not strictly taken into account in many of the previous work [49, 65, 92].

In our randomized distributed scheduling algorithm, the probability of random access is determined by queueing information gathered from each node's neighborhood. However, links make only infrequent local queue size information update, and use stale information for a long time of scheduling for random access This differs from previous techniques of using delayed information [124, 142] which require the queueing information, although delayed, to get updated in every time slot. While it is not known how the throughput region of their scheduling algorithms compare to the optimum throughput region with no information delay, we show that our approach based on infrequently updated delayed information can reduce the overhead of control information exchange to a negligible level and achieves a throughput region of $\Omega(1/g(L))\Lambda^{OPT}$, where Λ^{OPT} denotes the capacity region when the control information is not delayed.

2.3.4 Distributed Computing Model

Much of the traditional work on designing distributed algorithms [97,119] has focused on the CONGEST model of message passing, in which a node can communicate with all its neighbors in one time step. This model is more suited for wired networks, and the complexity of many fundamental problems (*e.g.*, MST [38], shortest paths [16], etc.) is very well understood in the wired model. This model does not capture *interference*, an inherent aspect of wireless networks, which causes collisions when "close-by" nodes transmit. The Radio Broadcast Network (RBN) Model (see, *e.g.*, [8,41]) was developed specifically to address such issues. It is known that the RBN model is significantly different from the CONGEST model [41,119], and its "local" structure has been used in designing efficient algorithms for many fundamental graph problems.

Though the RBN model is a much closer model of wireless transmission than the CONGEST model, it still does not capture several crucial features of SINR [47,102]: the SINR model more accurately captures the physical nature of wireless networks and has been used in a number of recent studies (*e.g.*, see the recent survey of Lotker and Peleg [95] and the references therein). However, unlike traditional (wired and RBN) models, in the SINR model distributed algorithms are especially challenging to design and analyze, because of the non-local nature of the model. In particular, to the best of our knowledge no prior distributed algorithms have been designed for "global" problems in the SINR model. Algorithms in the RBN model do not translate to efficient algorithms in the SINR model, whose non-locality makes it much harder.

In Chapters 8, we formally define the elements and list the assumptions for distributed computing in the SINR model.

2.3.5 Sensing-based Message-less Distributed Computing

We make crucial use of physical carrier sensing, and in solving MaxLSP-U and MaxLBP we let the wireless nodes make distributed decisions purely based on the Received Signal Strength Indication (RSSI) measurement without involving any message decoding. Given a threshold $Thres$, a node is able to detect if the total sensed power strength is $\geq Thres$. As discussed in [125], this can be done using the RSSI measurement possible through the Clear Channel Assessment capability in the 802.11 standard. In this way, the protocol is much simplified such that the wireless nodes only need to control the physical layer to access the medium with a certain power or to sense the channel.

2.3.6 Distributed Computing Sub-routine: (ω_1, ω_2)-ruling

An (ω_1, ω_2)-*ruling* $R_{\omega_1,\omega_2}(W)$ (where $\omega_1 < \omega_2$) (introduced in a graph model [31]) of a node set W, is a subset of nodes that covers the whole set of nodes while spatially separated. Formally, it has the following properties:

(1) $R_{\omega_1,\omega_2}(W) \subseteq W$;

(2) all the nodes in $R_{\omega_1,\omega_2}(W)$ are at least ω_1-separated; that is, $\forall u, u' \in R_{\omega_1,\omega_2}(W)$, $d(u, u') \geq \omega_1$; and

(3) W is ω_2-covered by $R_{\omega_1,\omega_2}(W)$; that is, $\forall u \in W, \exists u' \in R_{\omega_1,\omega_2}(W)$, $d(u, u') \leq \omega_2$

These properties make the nodes selected to be in the ruling separated from each other with a pre-specified distance and each of the other nodes is covered by one in the ruling within a pre-specified range in the network, and thus useful for many network topology control and connection control problems. We have developed and used the distributed (ω_1, ω_2)-ruling construction sub-routine for MaxLSP-U, MaxLBP; a variation of the (ω_1, ω_2)-ruling sub-routine has been developed and used for the non-uniform-power version of MaxLSP.

Chapter 3

Literature Review

In this chapter, we present an extensive review of the literature in the area of distributed scheduling and throughput-delay optimization for wireless networks.

3.1 Scheduling and Power Control with Dynamic Traffic

Despite the multitudes, the efforts on the problems with static traffic do not directly contribute to solving problems with dynamic traffic under physical interference model; and the problems with dynamic traffic under physical interference model still suffers from embarrassingly limited progress as summarized in Section 3.2.1.

3.1.1 Throughput Region

Since the seminal work of Tassiulas and Ephremides [132], based on the idea of maximal matching, a large number of algorithms are proposed, *e.g.*, variants of max-weight scheduling [49, 68], greedy maximal scheduling [64], maximal scheduling [27, 106]. Delay results are also established for the various algorithms as in 3.2; however, before our results, most of the end-to-end delay results are linear in network-size and the throughput-delay analysis under physical interference model is generally open.

As the scheduling complexity gets improved, the achievable throughput region usually gets traded-off. The back-pressure algorithm [132] performs routing and scheduling in a centralized fashion at the same time and attains optimum throughput region (which corresponds to the capacity region). While routes are fixed, the max-weight scheduling schemes pick a maximum-weighted independent set of links at each time slot as a valid schedule and secure an optimum throughput region; the maximal scheduling picks a maximal independent set of links at each time slot, and achieves $1/\mathcal{K}$ of the capacity region, where $\mathcal{K}$ is the interference degree, thus rendering an efficiency ratio of $1/\mathcal{K}$. There has been a flourishing amount of research effort on algorithm decentralization under graph-based models; Yet under physical interference model, notwithstanding, the distributed solution sees little progress due to the difficulties posed by interference aggregation and the non-convexity form of SINR.

3.1.2 Distributed Low-complexity Scheduling

In the past few years, much progress has been made in designing low-complexity distributed throughput-optimal scheduling schemes, most of which employ an *adaptive* approach, where the channel access parameters are dynamically adjusted from time to time (by referring to queue sizes).

A line of work uses adaptive random-access maximal scheduling coupled with a per-slot control phase in each slot, as in [23, 49, 65, 92], where knowledge of interference (or the conflict graph) is assumed. The control phase consists of multi-round message passing, during which contention resolution is realized by probabilistic channel requests and channel sensing. In [49], the algorithm Q-SCHED achieves a throughput region close to $\frac{1}{\mathcal{K}}\Lambda$, comparable to that of maximal scheduling. Follow-up work of [49] include [143] providing provable bounds on queue-overflow probability and [87] proposing a distributed algorithm with constant-time control phase under k-hop interference model.

A second line of work employs randomized policies based on a 'priori knowledge of traffic rates [26, 56]. These algorithms does not involve any message passing or queueing information. However, traffic information is not usually known. It is possible to estimate traffic rates with high accuracy for some stochastic arrival processes, while the estimation process may take a long time (*e.g.*, $O(e^n)$) for small rate values.

Another line of work is based on *adaptive CSMA*, where the channel occupation and back-off time parameters are adjusted according to local information (*i.e.*, queue size and transmission record). With elegant and rigorous proofs, the adaptive CSMA-based schemes are shown to be able to achieve optimal throughput region [63, 108, 121], even without message passing and per-slot control phase [94]. Nardelli et al. in [105] examine the performance of adaptive CSMA schemes and compare it with 802.11. They have shown that despite the promising theoretical performance, in practice, with hidden terminals and symmetric channels, the high aggressiveness of adaptive CSMA "further increases collisions yielding a self-sustaining loop" and "eventually, the protocol enters a state where no successful transmissions occur." In this case, "RTS/CTS does not help prevent the starvation." The large performance degradation of adaptive CSMA in practice is mainly due to the following aspects: (i) The assumption of time-scale separation between system dynamics and protocol behavior, where the system immediately converges to equilibrium is unrealistic [120]. (ii) No hidden nodes is a necessary condition and links are assumed to be load saturated (so that links always have useful packets to transmit; If links are allowed to be unsaturated, the explicit expression of the stationary distribution will be difficult). (iii) Starvation and fairness issues inherent to CSMA [34] are not resolved.

This suggests that interference information can be crucial in the performance of a scheduling scheme. While optimality can be achieved theoretically by adaptive CSMA, there could be some extreme cases in practice where it is hard to provide a lower bound on system performance. Therefore, we base our model on the knowledge of interference, namely, the conflict graph, which may be necessary for obtaining a performance lower bound in practice. It is also worth noting that even if interference information is known, without the knowing and sharing either the traffic rate information or the neighborhood queueing information, adaptive CSMA may still suffer from the same issues under a stochastic traffic model. Therefore, neighborhood queueing information is also a crucial factor.

3.2 Delay and Delay-constrained Throughput Capacity

Wired Networks: A large class of papers [10, 42, 89, 90, 110] provide analytical guarantees on the end-to-end delays and network utilization achievable through specific scheduling protocols in multi-hop *wired* networks. Parekh and Gallager [110] provide the first such per-session delay guarantees for weighted fair queueing protocol, followed by Georgiadis et al. [42] that present an analysis of EDF protocol. Then Andrews and Zhang [10], and Li and Knightly [89, 90] analyze a class of coordinated scheduling policies which decouple the path-length and rate dependencies of the delay. However, none of them explicitly deal with the problem of routing to simultaneously guarantee network utilization and end-to-end delays.

Table 3.1: Comparison of relevant scheduling and delay bound results for arbitrary wireless networks (for given traffic and flow routes with general interference). Notation used: n: #nodes; m: #links; $\mathcal{K}$: max interference degree; θ_{max}: max congestion (#flows through a link); $C(\mathcal{N})$: chromatic number of link interference graph.

Type	In Paper	Delay Bound	Scheduling Scheme
End-to-end	[59]	$O\left(\#hops \cdot D^2\right)$	preemptive LIFO, stable marriage
	[62]	$O\left(\#hops \cdot n\right)$	coordinated EDF or max-weight
	[86]	$O\left(\#hops_{max} \cdot \theta_{max} \cdot n \cdot m\right)$	max-weight
	[67]	$O\left((\#hops_{max})^2 \cdot C^{\mathcal{M}}(\mathcal{N})\right)$	max-weight indep. set
	[112] (Chapter 6)	$O\left((\#hops)^2\right)$	random access
	[50]	non-analytical	max-weight & variants
Single-hop	[106]	$O(1)$	maximal scheduling
	[66]	$O\left(C(\mathcal{N})\right)$	max-weight or random indep. set
Type	**In Paper**	**Efficiency Ratio**	**Type of Bound**
End-to-end	[59]	$\Omega\left(1/D^2\right)$	per-session, upper
	[62]	$\Theta(1)$	per-session, upper
	[86]	$\Theta(1)$	network-average, upper
	[67]	$\Theta(1)$	network-average, upper
	[112] (Chapter 6)	$\Omega\left(1/\mathcal{K}\right)$	per-session, upper
	[50]	$\Theta(1)$	network-average, lower
Single-hop	[106]	$\Omega\left(1/\mathcal{K}\right)$	network-average, upper
	[66]	$\Theta(1)$	network-average, upper

Random Wireless Networks: Precise tradeoffs between the network capacity and end-to-end delay as well as other parameters such as fairness, or number of radio channels (in a multi-radio multi-channel network) have been well studied for wireless networks under the assumption that the physical node locations follow *uniform spatial distributions*. Building on [48, 51], El Gamal et al. [39] show the relationship between average delay and (per-node) capacity. This problem has been extended in various directions, *e.g.*, [39, 107, 127, 128, 133]. However, in general, the techniques employed analyzing random wireless networks do not help shed light on the delay-throughput tradeoffs in an arbitrary wireless network (with non-random topology).

Arbitrary Wireless Networks: The design and analysis of wireless protocols for arbitrary networks (with non-random topologies) from the perspective of guaranteeing network utilization and end-to-end delays has received comparatively lesser focus [59, 62, 86]. The best bounds for end-to-end delays known before ours are by [59, 62, 67, 86]. Kar et al. [67] show that the average packet delay is bounded by the chromatic number $C(\mathcal{N})$ of the interference graph; $C(\mathcal{N})$ represents the minimum number of independent sets (# colors) into which the link conflict graph of a network $\mathcal{N}$ can be partitioned. Note that $C(\mathcal{N})$ is upper-bounded by $D+1$ (the maximum degree in the conflict graph) and is greater than $\mathcal{K}$ the maximum interference degree which is defined in Section 4; for example, in a "dense" network, $C(\mathcal{N})$ and D can be as high as $O(n)$. Jagabathula and Shah [59] design a scheduling scheme that ensures per-session end-to-end delays of $O\left(\#hops\right)$ with the total throughput within a constant factor of the optimum; however, this result is restricted to primary interference, whereas for general interference, the delay bound becomes $O\left(\#hops \cdot D^2\right)$, where D denotes the maximum degree in the conflict graph (which could be high). Jayachandran and Andrews [62] design a different scheduling scheme that ensures per-session end-to-end delay of $O\left(\#hops \cdot n\right)$. Le et al. [86] prove that max-weight scheduling has a network-average delay bound of $O\left(\#hops_{max} \cdot \theta_{max} \cdot n \cdot m\right)$. Thus, the bounds in prior results more or less depend on the network size; obtaining per-flow delay bounds independent of the network size had been an important open problem. Besides these end-to-end delay results, there are also important delay results for *single hop* traffic [50, 61, 66, 106]. Jaramillo, Srikant, Ying [61] propose a scheduling algorithm to satisfy long-term QOS requirements under heterogeneous delay constraints of a specific class of periodic traffic. Neely [106] shows that general maximal matching policies achieve $O(1)$ network-average delay for given traffic. Kar, Luo and Sarkar [66] show that the maximum expected delay depends linearly on the chromatic number of the interference graph. However, the one-hop traffic model is crucial for these results. Gupta and Shroff [50] give an algorithm for computing lower bounds for average delay under max-weight scheduling. We compare our scheduling and delay results with recent relevant results in Table 3.1. Our *unique* DCTM framework maximizes throughput with low per-session delay guarantees that only depend on target delays.

As for random-access policies, lots of efforts have been made to study end-to-end delay. However, most of them are constrained with using approximation models, with tools from statistical mechanics, such as diffusion-approximation model [22], totally asymmetric simple exclusion process [130], and approximate queueing model [123, 141]; it is not clear to what extent these simplified *approximate* statistical queueing models can be applied for the multi-hop dynamics in general networks with general arrival processes. We provide analytical bounds without simplifying either queueing or traffic model.

3.2.1 Physical Interference Model

There are very few existing work dealing with the scheduling problems with dynamic traffic under an SINR model, largely due to the complexity of SINR model and the difficulty of applying a Lyapunov function or fluid-limit approach. To our knowledge, throughput optimization problem can borrow solutions from the *weighted* one-shot scheduling problem, which targets to find the maximum weighted independent set of links that can be scheduled together. Goussevskaia, Oswald and Wattenhofer's centralized weighted one-shot scheduling algorithm [46] can be adapted and achieves a throughput region of $1/O(g(L))$ of Λ. Later, Goussevskaia, Halldórsson and Wattenhofer proposed an algorithm that solves non-weighted version of the one-shot scheduling problem with approximation ratio of $O(1)$ with a centralized algorithm [44]. These algorithms refer to optimality as the "constrained" optimality that is valid when subject to a given power assignment (*e.g.*, uniform power assignment). Kesselheim [69] provides the first $O(1)$-approximate algorithm for the non-weighted version of the one-shot scheduling problem with power control. These centralized algorithms may involve high overhead to be decentralized. Lee et al. [88], propose a power-control algorithm, which maximizes the throughput region, and which can be made decentralized but still involves expensive network-wise gossip and m-polynomial convergence time for scheduling at every time slot. They use a randomized approach that achieves a throughput region of $(1-\gamma_1)$ fraction of the optimal with probability $\Omega(\frac{1}{m^{1.5m}})$ where γ_1 is a parameter, and they do not consider a minimum required SINR for decoding. In this work, we present low-complexity distributed algorithms in a geometric SINR setting, and our approximation results refer to optimality that can be obtained by considering any scheduling and power assignment scheme.

3.3 Distributed Computing: MaxLSP, MaxLBP and MST

Distributed algorithms are known for distributed computing based on node and link scheduling (and many related problems) in the graph-based model [3, 4, 20, 96, 100, 126]. These algorithms are typically randomized and based on Luby's algorithm [96], and run in synchronous polylogarithmic time. There are varying assumptions on the kind of information and resources needed by individual nodes. For instance, [96] require node degrees at each step (which might vary, as nodes become inactive). Moscibroda et al. [100] develop algorithms that do not require the degree information, and run in $O(\log^2 n)$ time. Some results, *e.g.* [98, 126] use collision detection capabilities. In recent work, Afek et al. [4] develop a distributed algorithm for MaxLSP under graph-based interference model, with a constant approximation factor.

Distributed computing in the SINR model is considerably harder than in graph based models. Several papers developed $O(g(L))$-approximations for MaxLSP-U, *e.g.*, [25, 46], which have been improved to constant factor approximations by [44, 55, 137] for uniform power assignments. Some of these use "capacity" [44, 55] to refer to the maximum link scheduling; however, we prefer to avoid the term capacity in order to avoid confusion with the total throughput in a network, which has been traditionally referred to as the capacity (*e.g.*, [51]). Recently, Halldórsson and Mitra [53] extend the $O(1)$ approx. ratio to any setting with a fixed length-monotone, sub-linear power assignment (which includes uniform, mean and linear power assignments). This has been improved by Kesselheim [69], who developed the first $O(1)$-approximate scheduling and power control algorithm for the maximum independent set problem MaxLSP and an $O(\log m)$-algorithm for the minimum length scheduling problem, where the optimum refers to both scheduling and power control. Halldórsson and Mitra [53] propose a constant approximation algorithm for MaxLSP under a wide range of oblivious power assignments for both uni- and bi-directional links. Wan et al. [136] propose algorithms for a set of problems including maximum link scheduling and minimum length scheduling

Most of the above algorithms in the SINR model are centralized and and they do not provide clear clues of how to adapt them in a distributed manner efficiently. Few results are known before ours and the closest ones to ours are [12, 32]. The results of [12, 32] use a game theoretic approach which lead to much higher running times than ours.

A related set of problems of the MaxLSP and MaxLBP involves "coloring" in which the goal is to schedule *all* the transmission/broadcast requests in the smallest number of time slots, whereas in MaxLSP and MaxLBP we target to pick the largest feasible subset. Recently, for the "coloring" version of MaxLSP, Kesselheim and Vöcking [70] propose an $O(\log^2 m)$-approximate distributed algorithm for any fixed length-monotone and sub-linear power assignment. Its analysis has been improved to $O(\log m)$ by Halldórsson and Mitra [54], who also prove that if all links uses the same randomized strategy, there exists a lower-bound of $\Omega(\log m)$ on the running time. It is not clear how to use these results for the "coloring" problem to get a constant factor approximation for MaxLSP, in which the senders and receivers of all links know their status. Goussevskaia et al. [45] study the "coloring" version of MaxLBP (*i.e.*, to complete all the broadcast requests in shortest time). They design a sublinear-time random-access based distributed algorithm under the uniform power assignment and uniform broadcast range setting, depending on the information about the local topology. Nodes do not exchange information to determine who was successful. It is not clear how to adapt their algorithm for MaxLBP, especially with non-uniform ranges and the requirement that all nodes know whether they have been selected or not.

To our knowledge, we are the first to address the distributed problems — MaxLSP (except MaxLSP-U), MaxLBP and MST— under the physical interference model.

Chapter 4

System Model for Chapters 5, 6 and 7

Table 4.1: Notation for Chapters 5, 6 and 7.

G	network graph	n	#nodes
V	set of nodes	L	set of links
$\mathcal{K}$	interference degree	m	#links
$\mathcal{I}(l)$	interference set of l	$\hat{\mathcal{I}}(l)$	$\mathcal{I}(l) \cap \{l\}$
$x(l)$	transmitter of link l	$r(l)$	receiver of link l
$q_l(t)$	indicator: $Q_l(t) > 0$ at time t	$Q_l(t)$	backlog on l at time t
$d(l)$	length of link l	$d(u, v)$	dist. of u and v
$\mu(t)$	service rate at time t	$u(t)$	departure of Q at time t
λ	mean arrival rate	$A(t)$	exogenous arrival at time t
$\vec{\lambda}$	vector traffic rates	$\vec{Q}$	vector of backlogs
Λ	throughput region	Λ^{OPT}	capacity region
α	path-loss exponent	β	SINR threshold
N	background noise	$g(L)$	link diversity
$p_l(t)$	channel access prob.	Δ	delay constraint

4.1 Network Model

The wireless network is modeled as a directed graph $G = (V, L)$ in a Euclidean plane, where V is the set of nodes (or transceivers) and L the set of communication links. The network is synchronized and for simplicity we assume all slots have the same length. Links are directed, and for link $l = \big(x(l), r(l)\big)$, $x(l)$ and $r(l)$ denote the sender and the receiver, respectively. For nodes u and v, let $d(u,v)$ denote the Euclidean distance. For link l, let $d(l) \triangleq d(x(l), r(l))$ denote the length of the link; for links l, l', let $d(l', l) \triangleq d(x(l'), r(l))$. Each link l uses a transmission power level $P(l)$. We assume the commonly used path loss models [25, 46], in which the transmission on link l is possible only if:

$$\frac{P(l)/d^{\alpha}(l)}{N(1+\phi)} \geq \beta, \tag{4.1}$$

where $\alpha > 2$ is the "path-loss exponent," $\beta > 1$ is the minimum SINR required for successful reception, N is the background noise, and $\phi > 0$ is a constant (note that α, β, ϕ and N are all constants).

If not mentioned, we assume unidirectional channels, *i.e.*, at any time slot, communication is one way on each link. For the scheduling algorithms for data transmission under dynamic traffic to work in a distributed fashion, ACK messages may be necessary for confirming the reception of data packets at the same slot, and hence a bidirectional channel should be used. We will specify the use of either unidirectional or bidirectional channel model for each specific problem.

4.2 Wireless Interference

We study both graph-based interference model and the physical interference model based on geometric SINR constraints (henceforth also referred to as the SINR model):

(1) *Graph-based interference model.* As in [49, 106], interference is defined on a 2D binary symmetric adjacency matrix, such that the interference relationship between any two links is *binary* and *symmetric*. Each link l has its *interference sets* defined as $\mathcal{I}(l) = \{l' : l' \text{ interferes with } l\}$. This abstract interference model lacks specification of how links are connected, *e.g.*, transmission ranges; however, in order to communicate among interfering links, they need to be connected, either by infrastructure support (*e.g.*, enterprise WLANs) or by (intermediate) wireless links. For this reason, to account for the latter case we employ (but are not limited to) the widely used *k-hop interference model*: two links interfere with each other if and only if the shortest path between them on G is at most k hops. It is also possible to extend to other graph-based models. We consider *bidirectional* interference, which accounts for ACK packets that may be crucial in a distributed setting with random-access.

Hence, by l' *interferes with* l we mean that the transmission of either data from $x(l)$ or ACK from $r(l)$ will fail when either $x(l')$ or $r(l')$ is transmitting, or vice versa. This gives leeway to using asynchronous ACKs during a time slot. The *interference degree* $\mathcal{K} = \max_l \mathcal{K}(l)$, where $\mathcal{K}(l)$ is the maximum number of links that can make successful transmissions simultaneously in the interference set $\mathcal{I}(l)$. In most commonly used models of wireless interference, such as the distance-2 matching model [20], $\mathcal{K}$ is a constant. We assume that all the nodes have a common estimate of n, the size of the network, within a polynomial factor, and that each link knows the ID's of the links in its peripheral, *i.e.*, the ID's of the links in its interference set.

(2) *SINR interference model.* A subset $L' \subseteq L$ of links can be scheduled simultaneously if and only if the following condition holds for each $l \in L'$:

$$\frac{\frac{P(l)}{d^{\alpha}(l)}}{\sum_{l' \in L' \setminus \{l\}} \frac{P(l')}{d^{\alpha}(l',l)} + N} \geq \beta. \tag{4.2}$$

A *valid one-slot schedule* contains only links that can transmit successfully at the same time. We call a valid one-slot schedule an *independent set of links*. We say two (or more) links are *independent* of each other, if they together form an independent set.

4.3 Traffic and Queueing Models for Dynamic Traffic

4.3.1 Traffic Model

We assume the network is synchronized and time is divided into uniform and contiguous *slots* of unit length. Define $\mu_l(t) \in \{0, 1\}$ as the service rate for link l at time slot t; $\mu_l(t)$ is determined by the specific scheduling protocol used. For simplicity, we use time and link service models similar to [50, 66, 106]; our results can be extended to cases where link capacity is more than 1.

Let $\mathcal{C}$ denote the set of connections or sessions. Accordingly, we use $\mathcal{C}^{\checkmark}$ to denote the set of primary sessions, and $\mathcal{C}^{\int}$ to denote the set of secondary session requests. Let $s(c)$ and $t(c)$ denote the source and destination, respectively, for session $c \in \mathcal{C}$. Each session c might use multiple paths (also referred to as flows) for communication; let $\mathcal{F}(c)$ denote the set of paths/flows that can be used by session c. Let $L(f)$ denote the set of links on flow f. Let $A_f(t)$ denote the exogenous arrival process for flow f. We use i_f, where $i > 0$ is an integer, to denote the ith link on f (*e.g.*, 1_f is the 1st link in $L(f)$). We assume the exogenous arrival process of each flow to be i.i.d over time and independent of each other; the first moment $\mathbb{E}\{A_f(t)\} = \lambda(f)$ and the second moment $\mathbb{E}\{A_f^2(t)\} \le A^{(2)}$, where $A^{(2)}$ is a constant. Let $\lambda^p(l) = \sum_{c\in\mathcal{C}^p} \sum_{f\in\mathcal{F}(c):\ l\in L(f)}$ denote the total mean traffic rate of the flows that use link l in their paths.

In the setting of single-hop traffic, where flow paths boil down to single links, we replace flow and session symbols with link symbols, such that each link is associated with an exogenous arrival process $A_l(t)$, which is i.i.d. over time and independent of each other. The first moment, $\mathbb{E}\{A_l(t)\} = \lambda(l)$; the second moment, $\mathbb{E}\{A_l^2(t)\} \le A^{(2)}$, where $A^{(2)}$ is a constant. We further assume that $\exists A_{max} < \infty, A_l(t) < A_{max}, \forall l, \forall t$.

4.3.2 Queuing Model

In our queueing model, each link l is associated with a queue Q_l, which holds packets waiting for transmission. Q_l is divided into logical FIFO sub-queues for each flow through l. Let $Q_{l,f}$ denote the logical FIFO sub-queue for flow f on link l. We also use the same notation to denote the backlogs: For each time slot t, let $Q_l(t) \triangleq \sum_{f\in\mathcal{F}(l)} Q_{l,f}(t)$ denote the queue size, where $Q_{l,f}(t)$ denotes the number of packets waiting for transmission on link l for flow f, and where $\mathcal{F}(l)$ is the set of flows that include l in their paths*. Each time l is activated to transmit, only one logical queue on l gets serviced. Each packet of flow f traverses from $Q_{1_f,f}$ through $Q_{|L(f)|_f,f}$. For a slot t, let $a_{i_f,f}(t)$ be the number of arrival packets for f at $Q_{i_f,f}$, $u_{i_f,f}(t)$ the actual number of f's departure packets from $Q_{i_f,f}$, and $\mu_{i_f,f}(t)$ the service rate offered to $Q_{i_f,f}$. We employ a stop-and-wait ARQ mechanism. After data transmission at slot t, the transmitter node $x(l)$ will wait for an ACK from the receiver $r(l)$ for the rest of the slot. If no ACK is received until the end of t, $x(l)$ will deem the transmission as a fail so that $u_{i_f,f}(t) = 0$. We assume the transmission takes place during the entire slot, and at the end of the current slot, the arrival to a queue is counted in the backlog. The queue-evolution

*The notation used conforms to the following convention: for time-related quantities, subscripts are used to indicate specific links or flows, *e.g.*, Q_l, A_f; for non-time-related quantities, a function-like style is used to indicate specific links or flows, *e.g.*, $\mathcal{I}(l), L(f)$.

mechanism can be expressed as

$$\begin{aligned} Q_{i_f,f}(t+1) &= Q_{i_f,f}(t) - u_{i_f,f}(t) + a_{i_f,f}(t) \\ &= [Q_{i_f,f}(t) - \mu_{i_f,f}(t)]^+ + a_{i_f,f}(t), \end{aligned} \tag{4.3}$$

where $a_{i_f,f}(t) = \begin{cases} A_f(t), i = 1; \\ u_{(i-1)_f,f}(t), i = 2, 3, \ldots, |L(f)| \end{cases}$ and where $[a]^+ = \max\{0, a\}$.

For the single-hop setting, since each session corresponds to one link, we use the set of symbols $Q_l(t), u_l(t), \mu_l(t), a_l(t)$ to denote the queue size, departure rate, service rate, arrival rate respectively. The queue backlog evolves in the following manner:

$$Q_l(t+1) = Q_l(t) - u_l(t) + a_l(t).$$

4.3.3 Queue-Stability and Throughput Region

A schedule or scheduling scheme determines at each time t which links transmit. The system is said to be *queue-stable* (or strongly stable) [43] under a scheduling-power control scheme $\mathcal{SP}$ if and only if

$$\limsup_{t \to\infty} \frac{1}{t} \sum_{\tau \le t} \sum_{l \in L} \mathbb{E}\left\{Q_l(\tau)\right\} < \infty.$$

In the dissertation, we use queue-stable and stable interchangeably. The *throughput region* $\Lambda^{\mathcal{SP}}$ of a scheduling-power scheme $\mathcal{SP}$ is the closure of the set of all exogenous arrival rate vectors that can be stably supported under $\mathcal{SP}$. The *network capacity region* Λ^{OPT} is the closure of the set of all rate vectors that can be stably supported by any feasible scheduling scheme, which corresponds to optimum. It is known that the max-weight scheduling can stably schedule any traffic vector interior to Λ^{OPT} (but without any delay constraints) [43].

Theorem 4.1 (Necessary condition for stability)**.** *For a network to be stable, any traffic load needs to satisfy:* $\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \lambda(l') \le \mathcal{K}$, $\forall l \in L$, *where* $\mathcal{K}$ *is the interference degree (*i.e.*, the max. number of links that can make successful concurrent transmissions in any interference set).*

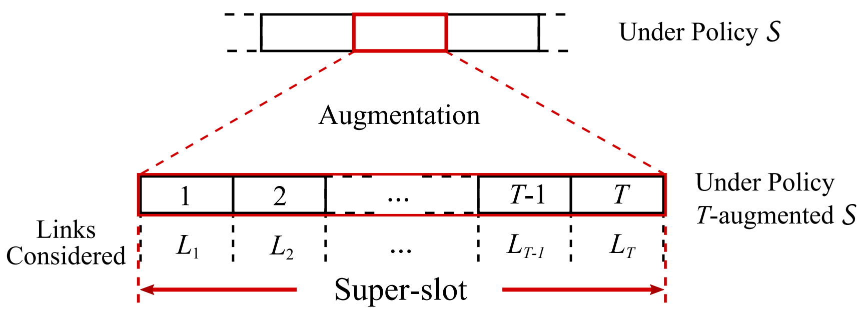


Figure 4.1: Structure of an augmented schedule.

4.4 SINR-specific Definitions

We introduce our new concepts and definitions related to the SINR model to aid our algorithm design and analysis. We partition the set L of links into length classes or link classes and define the link diversity $g(L)$, define interference set in the SINR setting, and show how to "augment" a schedule such that it can be adapted to link classes under the SINR model. Our approach to scheduling involves considering a time-slot t to be augmented into a set of time-slots and having only links of a given length class transmit in a given time-slot. This may scale down the efficiency ratio, but allows us to keep or reuse the general structure and analysis of algorithms developed with the graph-based interference model.

4.4.1 Link Diversity, Length Classes, Interference Set

Let $d_{min} = \min_{l \in L}\{d(l)\}$, and $d_{max} = \max_{l \in L}\{d(l)\}$. Let σ be a constant. We define *link diversity* $g(L) \triangleq \lceil \log_\sigma \frac{d_{max}}{d_{min}} \rceil$.

Partition $L = \{L_i\}, i = 1, 2, \ldots, g(L)$, where each $L_i = \{l \big| \sigma^{i-1} d_{min} \le d(l) < \sigma^i d_{min}\}$ is a *length class*, *i.e.*, the set of links of roughly similar lengths. Let $\mathcal{G}_i$ be the graph induced by the links in L_i. Let $d_i = \sigma^i d_{min}$. Let $\gamma = \sigma \sqrt[\alpha]{3 \cdot 2^3 \beta \frac{1+\phi}{\phi} \frac{\alpha-1}{\alpha-2}} + 2$. For link $l \in L_i$, we define $\mathcal{I}(l) \triangleq \{l' \big| l' \in L_i, d(l', l) \le \gamma d(l), \text{ or } d(l, l') \le \gamma d(l')\}$ as its *interference set*. This notion of interference set is symmetric.

4.4.2 Augmented Schedule/Scheduling

Given a scheduling policy $\mathcal{S}$ that does not discriminate link lengths, a T*-augmented* $\mathcal{S}$ is defined in the following manner (see Figure 4.1): under T-augmented $\mathcal{S}$, (i) each regular time slot t is augmented into a super-slot that consists of T slots, and (ii) in the ith slot of a super-slot, only links in length class L_i get scheduled using the same policy as $\mathcal{S}$.

4.4.3 Sub-linear Length-monotone Power Assignment

A power assignment is *sub-linear length-monotone* if $\frac{d(l)}{d(l')} \leq \frac{P(l)}{P(l')} \leq \frac{d^{\alpha}(l)}{d^{\alpha}(l')}$ whenever $d(l) \geq d(l')$. Intuitively, that is to say, the change (increase or decrease) rate of the transmission power on a link do not exceed the change rate of the signal fading due to spatial separation. One property of sub-linear length-monotone power assignments is that if $d(l)$ and $d(l')$ are within a constant factor of each other, then the ratio between $P(l)$ and $P(l')$ is (upper- and lower-) bounded by constants.

Chapter 5

Rate-oblivious Low-complexity Scheduling and Power Control for Wireless Networks

5.1 Preliminaries

The network model, traffic model, queueing model and related definitions are described in Chapter 4.

5.2 Problem Definition

In this chapter, we study the *throughput maximization problem*, formally defined as follows. Given an arbitrary wireless network, we make both *scheduling* and *power assignment* decisions to minimize a factor $C > 1$ so that for any traffic vector $\lambda \in \Lambda^{OPT}$, the network is stable under traffic vector $\frac{\lambda}{C}$; in other words, we maximize the throughput region. The factor C is the approximation factor, and $1/C$ is the efficiency ratio. In the chapter, we study the single-hop version of the problem. Our approximation ratio is based on considering an optimum of both scheduling and power control.

5.3 Algorithm for Graph-based Interference

We now describe RA-Sched in an abstract wireless interference model defined in terms of the interference sets ($\mathcal{I}(l)$ for each link l). RA-Sched uses queue length info to determine the random-access probabilities. RA-Sched runs in frames, and each frame has two sub-frames: an info-exchange sub-frame and a scheduling-tx sub-frame. RA-Sched allows the use of infrequently updated stale backlog information for scheduling, such that each link can operate on its own for a long time.

5.3.1 Description of Algorithm RA-Sched

Algorithm RA-Sched involves the following steps.

(1) **Framing**. Time is partitioned into contiguous frames of $H_0 = H_1 + H_2$ contiguous slots. A frame is further divided into two parts: an info-exchange sub-frame of H_1 slots and a scheduling-tx sub-frame of H_2 slots. In the *info-exchange sub-frame*, backlog info is exchanged among links and corresponding neighbors. Then in the *scheduling-tx sub-frame*, each link makes scheduling decisions based on the backlog info retrieved from the info-exchange sub-frame. Backlog info will not get updated during scheduling-tx.

(2) **Info-exchange**. Details of the backlog information exchange process are in Section 5.3.2. Let F be the current frame, and let τ be the time slot when F starts. Each link l maintains and updates its copies of values $V_{1,F}$ and $V_{2,F}$ for F, by using the backlog values at time τ:

$$V_{1,F}(l) = Q_l(\tau); \tag{5.1}$$

$$V_{2,F}(l) = \max_{l' \in \hat{\mathcal{I}}(l)} \sum_{l'' \in \hat{\mathcal{I}}(l')} Q_{l''}(\tau) = \max_{l' \in \hat{\mathcal{I}}(l)} \hat{Q}_{l'}(\tau). \tag{5.2}$$

Let ξ (addressed in Section 5.3.2) denote the probability that by the end of the info-exchange sub-frame, the info-exchange process is not completed, *i.e.*, at least one link has not fully updated backlog information for the current frame.

(3) **Random-access Scheduling-tx**. In each slot $t \in [\tau + H_1 + 1, \tau + H_0]$ of the scheduling-tx sub-frame, links use $V_{1,F}$ and $V_{2,F}$ values for scheduling decision. Let $q_l(t) \in \{0, 1\}$ indicate whether $Q_l(t) > 0$. Each link l attempts to access the channel with the following probability

at t.

$$p_l(t) = 1 - e^{-V_{1,F}(l)q_l(t)/V_{2,F}(l)}.$$

A collision happens when multiple links from the same neighborhood are activated at the same slot. Colliding transmissions are considered unsuccessful.

Algorithm 1: RA-SCHED

`/*` On each $l \in L$, do the following `*/`
while *a new frame* F **do**
  `/* in info-exchange sub-frame */`
  $V_{1,F}(l) \leftarrow Q_l(t_F(1))$;
  `InfoExchange`;
  $V_{2,F}(l) \leftarrow \max_{l' \in \hat{\mathcal{I}}(l)} \sum_{l'' \in \hat{\mathcal{I}}(l')} V_{1,F}(l'')$;
  repeat `/* in scheduling-tx sub-frame */`
    $p_l(t) \leftarrow 1 - e^{-V_{1,F}(l)q_l(t) \big/ V_{2,F}(l)}$;
    transmit with prob. $p_l(t)$ at a new slot t;
  until *end of frame* F;

5.3.2 Info-exchange Step

During this step, backlog information is exchanged among interfering links to update the value of $V_{2,F}(l)$ on each link l before scheduling-tx sub-frame of every frame F. We have the following two procedures.

(1) Multicast(L'): for every link $l \in L'$, its sender $x(l)$ successfully sends a message to the senders of all links in $\mathcal{I}(l)$, and
(2) LocalBroadcast(V'): every node $v \in V'$ successfully sends a message to all the nodes within one-hop distance.

Therefore, info-exchange is equivalent to two rounds of Multicast(L):

(1) in the first round of Multicast(L), each link l sends its $V_{1,F}(l)$ value to all the links in $\mathcal{I}(l)$, and
(2) in the second round of Multicast(L), each link l calculates and sends its $\sum_{l' \in \hat{\mathcal{I}}(l)} V_{1,F}(l')$ value to all the links in $\mathcal{I}(l)$ after l has collected all $V_{1,F}$ values from all the links in $\mathcal{I}(l)$.

Implementation of Multicast

Under k-hop interference model, we implement one round of Multicast(L) by $k{+}2$ invocations of LocalBroadcast(V), where data is also forwarded tagged with the ID of the origin link of the data. We use $(x(l), r(l))$ pair as the ID of a link l. For the first round of Multicast(L), each link l does the following:

(1) **Phase** 0. Let $X(L)$ denote the set of the senders of all the links in L. First, each node $v \in V$ creates a packaged message $M^{(0)}(v)$. One node may serve as a sender on multiple links. If $v = x(l)$ for any link $l \in L$, then $M^{(0)}(v)$ grows and includes the $V_{1,F}(l)$ information associated with the ID (*i.e.*, $(x(l), r(l))$ pair) of l. Then, we perform LocalBroadcast($X(L)$), such that for any link $l \in L$, the receiver $r(l)$ receives the $V_{1,F}(l)$ value, and $M^{(0)}(r(l))$ grows and includes the $V_{1,F}(l)$ information associated with the ID (*i.e.*, $(x(l), r(l))$ pair) of l.

(2) **Phase** $i \in [1, k]$. We perform LocalBroadcast(V), such that each node $v \in V$ successfully sends message $M^{(i-1)}(v)$ to all its one-hop neighbors. At the end of phase i each node v creates a new message $M^{(i)}(v)$ by merging all the messages it receives during the phase.

(3) **Phase** $k + 1$. We repeat phase 0 in a reverse direction, such that for any link $l \in L$, the sender $x(l)$ receives the $M^{(k)}(r(l))$ message from $r(l)$. The union of message $M^{(k)}(x(l))$ and $M^{(k)}(r(l))$ corresponds to the set of $V_{1,F}$ values from all the links in $\mathcal{I}(l)$, and their sum produces $V_{2,F}(l)$.

The second round of Multicast(L) involves the same $k + 2$ phases, except that we deal with $V_{2,F}$ values and calculate a local maximum in the end. The problem is hence reduced to given a set of nodes, making successful local broadcasts from each node to their one-hop neighbors in a distributed manner.

Implementation of LocalBroadcast

Procedure LocalBroadcast(V').

for $j = 1$ *to* $\frac{C \log n}{p_b}$ **do** `/*` $\frac{C \log n}{p_b}$ `slots in total */`
  each $v \in V'$ transmits with probability p_b;

The implementation of LocalBroadcast(V) is simple: Each node $v \in V$ transmits with probability p_b in each time slot for a period of length $\frac{C \log n}{p_b}$, where $C \geq 8$ is a constant. Let

$\mathcal{I}_b(v)$ of a node v denote the set of nodes, called *local broadcast interference set*, such that a local broadcast of v to all its one-hop neighbors is *successful* if and only if all the nodes in $\mathcal{I}_b(v)$ are silent. Let $BI \triangleq \max_{v\in V} |\mathcal{I}_b(v)|$ denote the maximum cardinality of the local broadcast interference sets. We discuss the value of p_b and show that LocalBroadcast(V) is completed in $\frac{C\log n}{p_b}$ time w.h.p. for the following two cases depending on the level of knowledge:

(1) Since each node has a common estimate of n, if we set $p_b = 1/n$, then at each time slot,

$$\begin{aligned}&Prob(\text{LocalBroadcast}(v) \text{ is completed})\\ =&p_b \prod_{u\in\mathcal{I}_b(v)} (1-p_b) \geq p_b(\frac{1}{4})^{\sum_{u\in\mathcal{I}_b(v)} p_b} \geq \frac{p_b}{4} = \frac{1}{4n}.\end{aligned}$$

After $Cn\log n$ slots, $Prob(\text{LocalBroadcast}(v) \text{ is not completed}) \leq (1-\frac{1}{4n})^{Cn\log n} \leq 1-\frac{1}{n^{C/4}}$. For LocalBroadcast$(V)$ to be completed (*i.e.*, all the n nodes has made successful local broadcast) in $Cn\log n$ slots, the probability is at least $1/n^{C/4-1}$.

(2) If each node v knows BI (which can be computed from the conflict graph), we set $p_b = \frac{1}{BI}$. The running time becomes $C\cdot BI\cdot\log n$. $Prob(\text{LocalBroadcast}(v) \text{ is completed}) \geq \frac{p_b}{4}$ at each time slot with the same analysis as the first case. We can also prove that LocalBroadcast(V) is completed w.h.p.

Since the size of a broadcast message is small, a regular slot can be divided to improve time cost. Hence, we have Lemma 5.1.

5.3.3 Analysis of Algorithm RA-Sched

We now analyze the performance of RA-Sched: the running time of each phase, and the throughput region.

Lemma 5.1. *If we set the length of the info-exchange sub-frame to $H_1 = \Theta(n\log n)$, then with prob. $\geq 1-\xi$, where $\xi \leq 2(k+2)/n^{C_0} \leq 1/n$ and $C_0 \geq 3$ is a constant, the $V_{1,f}$ and $V_{2,F}$ values can be updated successfully for all links by the end of info-exchange sub-frame. If each node knows BI, H_1 may be shortened to $\Theta(BI\cdot\log n)$.*

This lemma guarantees the info-exchange process is successful w.h.p. Rather pessimistically, failing to update $V_{1,f}$ and $V_{2,F}$ values on any link is considered to void an entire frame. That cost is fully taken into account in our analysis.

For proving stability and deriving throughput region, we use the idea of Lyapunov functions [43], which are a kind of potential functions. We define our Lyapunov function as

$$\mathbb{L}\big(\vec{Q}(t)\big) \triangleq \max_{k\in L} \sum_{k'\in\hat{\mathcal{I}}(k)} Q_{k'}(t) = \max_{k\in L} \hat{Q}_k(t),$$

where $\vec{Q}$ denotes the vector of queues.

Let F denote a frame and let $t_F(i)$ be the function that returns the time of the ith slot of F, *e.g.*, $t_F(1)$ is the time when F begins. Therefore, $V_{1,F}(l) = Q_l(t_F(1))$, and $V_{2,F}(l) = \max_{l'\in\hat{\mathcal{I}}(l)} \hat{Q}_{l'}(t_F(1))$. We define $V_{2,F}(L) \triangleq \mathbb{L}\big(\vec{Q}\,(t_F(1))\big)$, as the value of our Lyapunov function at time $t_F(1)$.

Intuitively, we first lower-bound how much can be sent from an interference set probabilistically when the global and local total values of backlogs exceed a certain value; then, we argue that the expected value of the Lyapunov function decreases w.h.p. in a certain number of frames; and finally, we prove that the system is stable under a certain condition on the traffic load, leading to the throughput region.

Lemma 5.2. *In a frame F, given the info-exchange sub-frame is successful,* RA-SCHED *guarantees that for any $\epsilon > 0$ and $h > 0, h < \infty$, if $V_{2,F}(L) \geq C_1/\epsilon$, then for any link l with $\sum_{l'\in\hat{\mathcal{I}}(l)} V_{1,F}(l') \geq V_{2,F}(L) - C_1 + H_2$, where $C_1 = 2hH_0\,|L|\,A_{max} + (h - h\epsilon - 2)H_2$, the following holds for any scheduling-tx slot $t \in [t_F(H_1+1), t_F(H_0)]$,*

$$\sum_{l'\in\hat{\mathcal{I}}(l)} Prob\{u_{l'}(t) = 1\} \geq (1-\epsilon)/e.$$

Lemma 5.3. *Let the current frame be F_0. If the average arrival satisfies $\forall l \in L, \sum_{l'\in\hat{\mathcal{I}}(l)} \lambda(l') \leq$*

$\frac{H_2}{H_0}(1 - 5\epsilon - \xi)/e,$

RA-SCHED *guarantees that for any constant $\epsilon' > 0$, there exists a positive finite number h that satisfies*

$\epsilon'/\,|L| \geq e^{-\frac{(hH_2\epsilon)^2}{2e^2}} + e^{-\frac{(h\epsilon)^2}{2e^2}} + e^{-\frac{2h(H_2\epsilon)^2}{H_0|L|^2A_{max}^2e^2}}$*, such that when $V_{2,F_0}(L) \geq C_1/\epsilon + (h-1)H_2$, the* *following holds for frame F_h, which is the hth frame after F_0:*

$$Prob\,\{V_{2,F_h}(L) \leq V_{2,F_0}(L) - hH_2\epsilon/e\} \geq 1 - \epsilon'. \tag{5.3}$$

The proofs of the above two lemmas are in Appendix. Lemma 5.3 says that there exists a finite number h, such that once the value of our Lyapunov function exceeds a finite number $C_1/\epsilon + (h-1)H_2$ at the start of a frame F, then in h frames, the expected value of the Lyapunov function decreases by at least a constant amount, w.h.p.

Theorem 5.4 (Sufficient condition of stability)**.** RA-SCHED *guarantees queue-stability with arrival processes satisfying*

$$\forall l \in L, \sum_{l' \in \hat{\mathcal{I}}(l)} \lambda(l') \leq \delta(1 - 5\epsilon - \xi), \forall \delta \in [0, 1/e).$$

Proof. Since H_1 is bounded by a finite number, for any positive constant $\delta < 1/e$, there exists a finite value for H_2, such that $\frac{H_2}{H_1+H_2} = e\delta$. Then, the arrival processes satisfy

$$\forall l \in L, \sum_{l' \in \hat{\mathcal{I}}(l)} \lambda(l') \leq \frac{H_2}{H_0}(1 - 5\epsilon - \xi)/e.$$

Recall our Lyapunov function $\mathbb{L}\big(\vec{Q}(t)\big) = \max_{l \in L} \sum_{l' \in \hat{\mathcal{I}}(l)} Q_{l'}(t)$. W.l.o.g., let t be the time when a frame starts.

Let J denote the event "$\mathbb{L}\big(\vec{Q}(t)\big) \geq C_1/\epsilon + (h-1)H_2$." From Lemma 5.3, we have: for any $\epsilon' < \frac{H_2\epsilon}{eH_0A_{max}|L|+H_2\epsilon}$, there exists a finite number h, s.t. when conditioned by J,

$$Prob\left\{\mathbb{L}\big(\vec{Q}(t+hH_0)\big) - \mathbb{L}\big(\vec{Q}(t)\big) \leq -hH_2\epsilon/e\right\} \geq 1 - \epsilon'.$$

That is saying, whenever J happens,

$$\begin{aligned}&\mathbb{E}\left\{\mathbb{L}\big(\vec{Q}(t+hH_0)\big) - \mathbb{L}\big(\vec{Q}(t)\big) \mid \vec{Q}(t)\right\}\\ \leq& -(1-\epsilon')hH_2\epsilon/e + hH_0A_{max}\,|L|\,\epsilon' < 0.\end{aligned}$$

By Foster's theorem [43], the system is queue-stable. □

Theorem 4.1 and 5.4 together characterize a throughput region of almost $\frac{1}{e\mathcal{K}}\Lambda^{OPT}$. The efficiency ratio of RA-SCHED is therefore approaching $\frac{1}{e\mathcal{K}}$. In the proof of Theorem 5.4, through $\mathcal{K}$, it suggests the relation between the choice of the value of H_2 and the throughput region. The larger H_2 is, the closer the throughput region is to $\frac{1}{e\mathcal{K}}\Lambda^{OPT}$.

5.4 Description of Algorithm RA-Sched-SINR

(Recall the SINR-specific definitions in Chapter 4.) RA-SCHED-SINR is a $g(L)$-augmented version of RA-SCHED over link set L for scheduling in SINR setting, with the modifications listed below this paragraph. A slot under RA-SCHED extends to a super-slot under RA-SCHED-SINR, where RA-SCHED is performed over links in L_i at the ith slot of a super-slot. As a result of the augmentation, the length of a frame is enlarged by a factor of $g(L)$. Let $H_1' = g(L)H_1$ and $H_2' = g(L)H_2$ denote respectively the lengths of the two sub-frames of a frame for RA-SCHED-SINR. Let $H_0' = H_1' + H_2'$ denote the length of a frame. In each frame each link set L_i is allocated H_1 slots for info-exchange, and H_2 slots for scheduling-tx. Below lists our modifications based on $g(L)$-augmented RA-SCHED (recall definitions in Section 5.3.1):

(1) **Channel-access Probability.** We reset the probability of accessing the channel for any link l to

$$p_l(t) = 1 - e^{-V_{1,F}(l)q_l(t)/\left(\Gamma V_{2,F}(l)\right)},$$

where $\Gamma > 1$ is a constant parameter whose value will be determined later (in Section 5.4.1) to achieve optimal effect.

(2) **Local Broadcast Interference Set.** We redefine the *local broadcast interference set* in SINR setting for info-exchange as: for each node $v \in X(L_i)$ (where $X(L_i)$ is the set of the senders of all the links in L_i) in the ith slot of a super-slot,

$$\mathcal{I}_b(v,i) \triangleq \left\{u \middle| \min\{d(u,v)\} \leq 2\gamma d_i, \forall u \in X(L_i)\right\}.$$

Let $BI \triangleq \max_{i\in[1,g(L)]} \max_{v\in V} |\mathcal{I}_b(v,i)|$ denote the maximum cardinality of the local broadcast interference sets.

(3) **Power Assignment.** The only constraint on power assignment is that, the power values for transmission on any two links in the same set L_i are at most a constant factor away from

each other. This incorporates uniform and linear power assignment schemes as special cases. For simplicity, we assume links in the same link set L_i use the same transmission power P_i. Removing this assumption may affect the results by no more than a constant factor.

(4) **Backlog Info-exchange.** Each link l boosts up the power level by a constant factor, such that the transmission range can reach the transmitter of any link in $\mathcal{I}(l)$ (a circular region). The info-exchange sub-frame consists of two equi-length rounds of local broadcast. In the first round, each link l sends the $V_{1,F}(l)$ value to $\mathcal{I}(l)$, and in the second round sends the value of $\sum_{l' \in \hat{\mathcal{I}}(l)} V_{1,F}(l')$ to $\mathcal{I}(l)$ if l has collected all $V_{1,F}(l')$ values from each $l' \in \mathcal{I}(l)$.

5.4.1 Correctness and Analysis of RA-Sched-SINR

Theorem 5.5 gives the conditions under which RA-SCHED-SINR is stable; it follows that our algorithm has an efficiency ratio of $\frac{1}{O(g(L))}$.

Theorem 5.5 (Sufficient condition for stability). RA-SCHED-SINR *guarantees queue-stability of the system under arrival processes that satisfy*

$$\forall l \in L, \sum_{l' \in \hat{\mathcal{I}}(l)} \lambda(l') \leq \delta(1 - 5\epsilon - \xi), \tag{5.4}$$

for any positive constant $\delta < \frac{1}{g(L)} \frac{1-1/\Gamma}{e^{1/\Gamma}\Gamma}$*, where* $\Gamma = e$*.*

To prove Theorem 5.5, we start with the analysis of each phase, which provides insights to the algorithm.

Lemma 5.6. *For a choice of* H_1' *such that* $H_1' = \Theta\big(g(L) n \log n\big)$*, the* $V_{1,F}$ *and* $V_{2,F}$ *values are updated successfully on all links by the end of an info-exchange sub-frame, w.h.p. If each node knows* BI*,* H_1 *may be shortened to* $\Theta\left(g(L) \cdot BI \cdot \log n\right)$*.*

(sketch). The proof of the performance builds on the results from [45]. Each node needs to know when the info-exchange sub-frame ends to enter the next sub-frame. The running time of info-exchange needs to be based on some common knowledge — n or BI — resulting in different time length.

Goussevskaia, Moscibroda, and Wattenhofer [45] propose two distributed randomized local broadcasting algorithms for each node to broadcast to all nodes in its proximity with a radius of R_B. Their first algorithm requires the knowledge of the number of nodes within a distance of $R_A \geq 2R_B$. Let $\mathcal{K}_A$ denote the maximum number of nodes at most R_A away from any node. By letting each node transmits with probability $1/\mathcal{K}_A$ for $\Omega\left(\mathcal{K}_A \log n\right)$ time, they show that all the local broadcasts can finish with probability of at least $1 - 1/n^C$, where n is the number of nodes and C is a constant.

In our SINR model, we treat the senders of all links as nodes that participate in the local broadcasts. We set $R_A = 2\gamma d_i$ and $R_B = \gamma d_i$ so that $R_A \geq R_B$ and R_B covers all the target nodes of the local broadcast of a node. Only difference is that each node makes two rounds of broadcasts (for value $V_{1,F}$ and $V_{2,F}$ respectively). By using the result in [45], we obtain a probability guarantee of at least $1 - 2/n^C$ and $O\left(g(L) \cdot BI \cdot \log n\right)$ time for all links to finish info-exchange. When the knowledge of BI is not available, if each node transmits with probability $1/n$, in $O\left(g(L) n \log n\right)$ time all the links can finish info-exchange w.h.p. □

Proposition 5.7. *The expected number of active (or transmitting) links in $\hat{\mathcal{I}}(l)$ of any $l \in L$ at any time t is $\leq 1/\Gamma$.*

Proposition 5.8. *We say two links k and k' are "independent" if they are not in each other's interference set. The distance between either end of k and either end of k' is at least $(\gamma - 2)d_i$.*

Lemma 5.9. *Let t be a time slot when links from L_i are scheduled. The probability for any $l \in L_i$ to make successful transmission when there is no interference from $\mathcal{I}(l)$ is*

$$Prob\Big(u_l(t) = 1 \Big| \sum_{l' \in \mathcal{I}(l)} s_{l'}(t) = 0\Big) \geq p_l(t)(1 - 1/\Gamma).$$

Proof. We first partition the plane into concentric rings via a similar technique to that in [25, 103], and then by solving a set cover problem we develop an upper-bound on the minimum number of greater interference sets ($\hat{\mathcal{I}}$) that covers all the links that touches a ring, so that with reference to Proposition 5.7, we can bound the expected total interference from all the rings outside of $\mathcal{I}(l)$.

We partition the plane into rings all centered at $r(l)$, each of width $(\gamma - 2)d_i$. Let $R(j)$ denote the jth ring, which contains every transmitter node u of a link in L_i, that satisfy $j(\gamma - 2)d_i \leq d(u, r(l)) < (j+1)(\gamma - 2)d_i$, where $j = 1, 2, \ldots$. When $j = 0$, $R(0)$ corresponds to a disk of radius $(\gamma - 2)d_i$, within which the nodes all belong to links in $\hat{\mathcal{I}}(l)$ due to Proposition 5.8. We say that link $k \in R(j)$, if either $x(k)$ or $r(k)$ falls in $R(j)$. Let the total number of active (or transmitting) nodes in $R(j)$ at t be $N_{R(j)}(t)$. Since each link attempts to access the channel and transmit independently, and each link can have at most one active node (either sending data from transmitter node or sending ACK from receiver node), $N_{R(j)}(t)$ may fluctuate during t; yet, we have that during t, $\mathbb{E}\{N_{R(j)}(t)\} \leq \sum_{k\in R(j)} p_k(t)$.

Let $L^{cov}(j)$ be a minimal set of links in $R(j)$, such that

(1) $\bigcup_{k\in L^{cov}(j)} \hat{\mathcal{I}}(k)$ covers all the links in $R(j)$, and

(2) all the links in L^{cov} do not interfere with each other.

By Proposition 5.7,

$$\mathbb{E}\{N_{R(j)}(t)\} \leq \sum_{k\in L^{cov}(j)} \sum_{k'\in\hat{\mathcal{I}}(k)} p_{k'}(t) \leq |L^{cov}(j)|/\Gamma. \tag{5.5}$$

$L^{cov}(j)$ is a set of pairwise independent links in $R(j)$. Now we pick a node set $N^{cov}(j)$ that covers all links in $L^{cov}(j)$, such that each link has one and only one node in $N^{cov}(j)$, and all the nodes in $N^{cov}(j)$ fall in $R(j)$. By Proposition 5.8, disks of radius $(\gamma-2)d_i/2$ centered at each of the nodes in $N^{cov}(j)$ do not overlap. All these disks are contained in an extended ring $R'(j)$ of $R(j)$, with extra width of $(\gamma-2)d_i/2$ at each side of $R(j)$.

Therefore, $|L^{cov}(j)| = |N^{cov}(j)| \leq 2^3(2j+1)$ due to packing property. That yields from Inequality (5.5) that

$$\mathbb{E}\{N_{R(j)}(t)\} \leq 2^3(2j+1)/\Gamma.$$

Since on some links it may be the transmitter nodes that are transmitting data and on others it may be the receiver node that are transmitting ACKs, the interference observed can vary in a time slot. Let $I^x_j(t)$ and $I^r_j(t)$ denote the maximum total interference from the jth ring to $x(l)$ and $r(l)$ respectively at t. By noticing that $x(l)$ is on the circle centered at $r(l)$ with radius $d(l)$, we have

$$\begin{cases} I^x_j(t) \leq N_{R(j)}(t)\cdot \frac{P_i}{(j(\gamma-2)d_i-d(l))^\alpha}; \text{ and} \\ I^r_j(t) \leq N_{R(j)}(t)\cdot \frac{P_i}{(j(\gamma-2)d_i)^\alpha}. \end{cases}$$

Then, define $I_j(t) \triangleq N_{R(j)}(t)\cdot \frac{P_i}{(j(\gamma-3)d_i)^\alpha}$. Therefore, $I^x_j(t) \leq I_j(t)$ and $I^r_j(t) \leq I_j(t)$ at the same time.

Let $I^x(t) \triangleq \sum_{j=1}^\infty I^x_j(t)$ and $I^r(t) \triangleq \sum_{j=1}^\infty I^r_j(t)$ denote the maximum total interference from outside of $\mathcal{I}(l)$ respectively. Let $I(t) = \sum_{j=1}^{\infty} I_j(t)$; then, $I(t) \geq \max\{I^x(t), I^r(t)\}$.

$$\mathbb{E}\{I(t)\} \leq \mathbb{E}\{\sum_{j=1}^{\infty} I_j(t)\} = \sum_{j=1}^{\infty} \mathbb{E}\{I_j(t)\}$$
$$= \sum_{j=1}^{\infty} \frac{2^3}{\Gamma} P_i(\gamma-3)^{-\alpha} d_i^{-\alpha} \frac{2j+1}{j^\alpha}$$
$$\leq \frac{3\cdot 2^3}{\Gamma} P_i(\gamma-3)^{-\alpha} d_i^{-\alpha} \frac{\alpha-1}{\alpha-2}.$$

Recall that $s_l(t) \in \{0,1\}$ indicates whether link l chooses to access the channel at time t. Let J_1 denote the event that "$s_l(t)=1$," J_2 the event that "$\sum_{l' \in \mathcal{I}(l)} s_{l'}(t) = 0$," and J_3 the event that "$I(t) < \Gamma \cdot \mathbb{E}\{I(t)\}$." According to Markov's Inequality:

$$Prob\,(J_3) = 1 - Prob\big(I(t) \geq \Gamma \cdot \mathbb{E}\{I(t)\}\big) \geq 1 - 1/\Gamma.$$

With J_1, J_2, and J_3 being true at the same time, the SINR values at $x(l)$ and $r(l)$ are at least

$$\frac{P_i d^{-\alpha}(l)}{\Gamma\mathbb{E}\{I(t)\}+N} \geq \frac{P_i d_i^{-\alpha}}{3\cdot 2^3 \sigma^\alpha P_i (\gamma-3)^{-\alpha} d_i^{-\alpha}\frac{\alpha-1}{\alpha-2} + N} \geq \beta.$$

That means both data transmission from $x(l)$ and ACK transmission from $r(l)$ will be successful, *i.e.*, $u_l(t) = 1$.

Therefore, $Prob\,\big(u_l(t)=1 \big| J_2\big) \geq Prob\,\big(J_1 \cap J_3 \big| J_2\big)$ leading to the statement. □

We define a Lyapunov function as before in Section 5.3.3:

$$\mathbb{L}\left(\vec{Q}(t)\right) \triangleq \max_{k\in L} \sum_{k' \in \hat{\mathcal{I}}(k)} Q_{k'}(t) = \max_{k \in L} \hat{Q}_k(t).$$

In the same way, let F denote a frame and let $t_F(i)$ be the function that returns the time of the ith slot of F. Therefore, $V_{1,F}(l) = Q_l(t_F(1))$ and $V_{2,F}(l) = \max_{l' \in \hat{\mathcal{I}}(l)} \hat{Q}_{l'}(t_F(1))$. We use $V_{2,F}(L) = \mathbb{L}\big(\vec{Q}\,(t_F(1))\big)$ to denote the value of our Lyapunov function at $t_F(1)$.

Lemma 5.10. *In a frame F, given the info-exchange sub-frame is successful,* RA-Sched-SINR *guarantees that for any $\epsilon > 0$ and $h > 0, h < \infty$, if $V_{2,F}(L) \geq C_1/\epsilon$, then for any i and any link $l \in L_i$ with $\sum_{l' \in \hat{\mathcal{I}}(l)} V_{1,F}(l') \geq V_{2,F}(L) - C_1 + H_2$, where $C_1 = 2hH_0'|L|A_{max} +$*

$(h - h\epsilon - 2)H_2$, the following holds in the scheduling-tx sub-frame:

$$\sum_{l' \in \hat{\mathcal{I}}(l)} Prob\{u_{l'}(t) = 1\} \geq \frac{(1 - 1/\Gamma)}{e^{1/\Gamma}\Gamma}(1 - \epsilon), \forall i,$$
$$\forall t = t_F(H_1' + i + jg(L)), \text{ where } j = 0, 1, 2, \ldots, H_2 - 1.$$

Proof. (sketch) W.l.o.g., we look at a slot $t = t_F(H_1' + i + jg(L))$, when only links from L_i are being scheduled. There are in total $H_2 = H_2'/g(L)$ such slots in frame F. Let $l \in L_i$ be a link with $\sum_{l' \in \hat{\mathcal{I}}(l)} V_{1,F}(l') \geq V_{2,F}(L) - C_1 + H_2$. Let J_4 denote the event that "$\sum_{k' \in \mathcal{I}(k)} s_{k'}(t) = 0$." By applying Lemma 5.9, any link $k \in \mathcal{I}(l)$ makes a successful transmission with probability

$$\begin{aligned} &Prob\{u_k(t) = 1\} \geq Prob\left(u_k(t) = 1 \cap J_4\right) \\ \geq &Prob\left(u_k(t) = 1 \;\middle|\; J_4\right) \cdot Prob(J_4) \\ \geq &\left(e^{\frac{V_{1,F}(k)q_k(t)}{\Gamma V_{2,F}(L)}} - 1\right)\Big/ e^{1/\Gamma} \geq \frac{(1 - 1/\Gamma)V_{1,F}(k)q_k(t)}{e^{1/\Gamma}\Gamma V_{2,F}(L)}. \end{aligned}$$

Then, using the same reasoning in the proof of Lemma 5.2 will complete the proof. □

Following the above lemma which is equivalent to Lemma 5.2, we have Lemma 5.11 below equivalent to Lemma 5.3.

Lemma 5.11. *Let the current frame be F_0. If the average arrival satisfies for any $l \in L$,*

$$\sum_{l' \in \mathcal{I}\hat{(l)}} \lambda(l') \leq \frac{H_2'}{H_0' g(L)}(1 - 5\epsilon - \xi)\frac{(1 - 1/\Gamma)}{e^{1/\Gamma}\Gamma},$$

where $\Gamma = e$ (discussed in Section 5.4.1), RA-Sched-SINR *guarantees that for any constant $\epsilon' > 0$, there exists a finite number $h > 0$ that satisfies $\epsilon'/|L| \geq e^{-\frac{(hH_2\epsilon)^2}{2e^2}} + e^{-\frac{(h\epsilon)^2}{2e^2}} + e^{-\frac{2h(H_2\epsilon)^2}{H_0'|L|^2 A_{max}^2 e^2}}$, such that when we have $V_{2,F_0}(L) \geq C_1/\epsilon + (h-1)H_2$, the following holds for frame F_h, which is the hth frame after F_0:*

$$Prob\left\{\sum_{l' \in \hat{\mathcal{I}}(l)} V_{2,F_h}(L) \leq V_{2,F_0}(L) - hH_2\epsilon/e\right\} \geq 1 - \epsilon'.$$

(sketch). Whether "$\sum_{l' \in \hat{\mathcal{I}}(l)} Q_{l'}(t_F(1)) \leq$
$V_{2,F}(L) - hH_0'|L|A_{max} - hH_2\epsilon/e$" is true or not gives us two cases. After applying similar

techniques in Lemma 5.3, the statement follows. □

Theorem 5.5 can be then proved in the same way as Theorem 5.4.

Determining the value of Γ

We notice that $f(x) = \frac{1-1/x}{e^{1/x}x}$ is not a monotone function of $x > 1$. Hence, we would like to find a value of Γ to maximize $f(\Gamma)$, such that the limit of δ in Theorem 5.5 and the underlying throughput region get as large as possible. By solving "$\max_{x>1} \frac{1-1/x}{e^{1/x}x}$" we can find a solution x^*, where $2.5 < x^* \approx e < 2.8$. We further notice that $|f(x^*) - f(e)| \leq 0.001$. Simply taking $\Gamma = e$ yields $\dfrac{(1-1/\Gamma)}{e^{1/\Gamma}\Gamma} = \dfrac{1-1/e}{e^{1+1/e}}$.

Efficiency Ratio of RA-Sched-SINR

For RA-Sched-SINR, Theorem 5.5 and Theorem 4.1 together characterize a throughput region of nearly $\frac{1}{\mathcal{K}}\frac{(1-1/e)}{g(L)e^{1+1/e}}$ of the optimum Λ^{OPT}. $\mathcal{K}$ in the SINR setting is the cardinality of a maximum independent set under optimum power assignment in $\mathcal{I}(l)$ of any link l. Due to Lemma 5.12 (whose proof is in Appendix), $\mathcal{K}$ is at most a constant. We conclude that the efficiency ratio of RA-Sched-SINR reaches $\Omega(1/\,g(L))$.

Lemma 5.12. *Under any scheduling and power assignment scheme, $\mathcal{K}$ is a at most a constant in our SINR setting.*

5.5 Chapter Summary and Discussion

In this chapter, we develop the first local distributed scheduling algorithm (RA-Sched-SINR) that provably ensures a throughput region of $\Omega(\frac{1}{g(L)})\Lambda^{OPT}$ in the SINR model, with Λ^{OPT} being optimum considering both scheduling and power control. We also design RA-Sched, a distributed scheduling scheme with graph-based interference, which forms the basis of RA-Sched-SINR. Our algorithms are based on random access, and use local queue size info for scheduling. We rigorously analyze all aspects of communication complexity, including information update and exchange, whose frequency can be reduced, to limit the overall overhead.

One application of RA-SCHED and RA-SCHED-SINR is in WLANs where local info-exchange is easy. Consider the downlinks scheduling problem in [58, 129], with WLANs consisting of multiple Access Points (or APs) which are wired with access to the Internet. Downlink traffic is reported to constitute as much as 80% of the total traffic in enterprise WLANs [129]. Because in such a setting it is easier to obtain the conflict graph — *e.g.*, a micro-probing approach [5] for online conflict graph construction can obtain contention information within millisecond time scales [58, 129] — the time for info-exchange step can be improved (so that the delay performance improves) based on that. Since the length of a slot is at the level of tens to hundreds of milliseconds, our algorithms may serve as simple distributed solutions with low overhead.

5.6 Appendix to Chapter 5

5.6.1 Proof of Lemma 5.2

Let k be a link in $\hat{\mathcal{I}}_l$. The probability for k to make a successful transmission at time t is:

$$\begin{aligned} &Prob\{u_k(t)=1\} = p_k(t)\prod_{k'\in\mathcal{I}_k}(1-p_{k'}(t)) \\ \geq&(1-e^{-\frac{V_{1,F}(k)q_k(t)}{V_{2,F}(L)}})\, e^{-\frac{\sum_{k'\in\mathcal{I}_k\cup\{k\}}V_{1,F}(k')q_{k'}(t)-V_{1,F}(k)q_k(t)}{\sum_{k'\in\mathcal{I}_k\cup\{k\}}V_{1,F}(k')q_{k'}(t)}} \\ \geq&(e^{\frac{V_{1,F}(k)q_k(t)}{V_{2,F}(L)}}-1)/e \geq \frac{V_{1,F}(k)q_k(t)}{eV_{2,F}(L)}. \end{aligned}$$

Now let us look at link l, with $\sum_{l'\in\hat{\mathcal{I}}_l}V_{1,F}(l') \geq V_{2,F}(L)-C_1+H_2$. Because at most H_2 packets can be transmitted from all the links in $\hat{\mathcal{I}}_l=\mathcal{I}_k\cup\{k\}$ during H_2 slots of the scheduling-tx sub-frame, we have $\sum_{l'\in\hat{\mathcal{I}}_l}V_{1,F}(l')q_{l'}(t)\geq V_{2,F}(L)-C_1$.

Therefore, $\displaystyle\sum_{l'\in\hat{\mathcal{I}}_l}Prob\{u_{l'}(t)=1\}\geq\sum_{l'\in\hat{\mathcal{I}}_l}\frac{V_{1,F}(l)q_l(t)}{eV_{2,F}(L)}\geq\sum_{l'\in\hat{\mathcal{I}}_l}\frac{V_{2,F}(L)-C_1}{eV_{2,F}(L)}\geq(1-\epsilon)/e.$ $\square$

5.6.2 Proof of Lemma 5.3

Let l be an arbitrary link in L. The total amount of arrival packets within any interference set is bounded as

$$\sum_{l' \in \hat{\mathcal{I}}_l} A_{l'}(t) \leq |L| \, A_{max}, \forall t, \forall l.$$

For any link l, there are two cases as below.

Case 1: $\sum_{l' \in \hat{\mathcal{I}}_l} Q_{l'}(t_F(1)) \leq V_{2,F}(L) - hH_0 \, |L| \, A_{max} - hH_2 \epsilon / e$.

Since in total at most $hH_0 \, |L| \, A_{max}$ new packets arrives at all links in $\hat{\mathcal{I}}_l$, we have

$$Prob\Big\{ \sum_{l' \in \hat{\mathcal{I}}_l} Q_{l'}(t_{F_0}(1) + hH_0) \leq V_{2,F_0}(L) - hH_2 \epsilon / e \Big\} = 1.$$

Case 2: $\sum_{l' \in \hat{\mathcal{I}}_l} Q_{l'}(t_F(1)) > V_{2,F}(L) - hH_0 \, |L| \, A_{max} - hH_2 \epsilon / e$.

We first assume that the info-exchange process is always successful, and later we will remove this assumption. Let F_i be any one of the h frames before frame F_h starts, *i.e.*, $i = 0, 1, 2, \ldots, h-1$. Because at most one packet can be transmitted from links in $\hat{\mathcal{I}}_l$, we have $V_{2,F_0}(L) + hH_0 \, |L| \, A_{max} \geq V_{2,F_i}(L) \geq V_{2,F}(L) - (h-1)H_2 \geq C_1/\epsilon$. Hence,

$$\begin{aligned} \sum_{l' \in \hat{\mathcal{I}}_l} V_{1,F_i}(l') &= \sum_{l' \in \hat{\mathcal{I}}_l} Q_{l'}(t_{F_i}(1)) \\ &\geq \sum_{l' \in \hat{\mathcal{I}}_l} Q_{l'}(t_F(1)) - (h-1)H_2 \geq V_{2,F_i}(L) - C_1 + H_2. \end{aligned}$$

Therefore, we have $\sum_{l' \in \hat{\mathcal{I}}_l} Prob\,\{u_{l'}(t) = 1\} \geq (1-\epsilon)/e, \forall t \in [t_{F_i}(H_1+1), t_{F_i}(H_0)], \forall i \leq h-1$, according to Lemma 5.2. Then, let $u_{\hat{\mathcal{I}}_l}(t) \in \{0, 1\}$ indicate whether there is a packet transmitted from any link among $\hat{\mathcal{I}}_l$; we have

$$\mathbb{E}\left\{u_{\hat{\mathcal{I}}_l}(t)\right\} \geq (1-\epsilon)/e, \forall t \in [t_{F_i}(H_1+1), t_{F_i}(H_0)], \forall i \leq h-1.$$

Now we remove the assumption that in all the h frames, all the info-exchange processes are successful. Considering the probability of a failed info-exchange process, let Z denote the total number of frames with successful info-exchange out of the h frames; $Z = \sum_{i=0}^{h-1} z(i)$, where $z(i) \in \{0,1\}$ indicates whether the info-exchange process is successful in frame F_i. $\mathbb{E}\{z(i)\} \geq 1-\xi$, $\forall i \leq h-1$. Using the "lower tail" Chernoff's Bound yields $Prob\{Z \geq h(1-\xi-\epsilon)\} \geq 1 - e^{-\frac{(h\epsilon)^2}{2e^2}}$.

W.l.o.g., we assume $h(1-\xi-\epsilon)$ to be an integer. Let X denote the total amount of transmitted packets in h frames; $X = \sum_{t=t_{F_0}(1)}^{t_{F_0}(1)+hH_0-1} u_{\hat{\mathcal{I}}_l}(t)$. Applying the "lower tail" Chernoff Bound again gives us the following conditional probability

$$Prob\left\{X \geq ZH_2(1-2\epsilon)/e \,\middle|\, Z \geq h(1-\xi-\epsilon)\right\} \geq \left(1 - e^{-\frac{(hH_2\epsilon)^2}{2e^2}}\right).$$

Therefore, we have

$$\begin{aligned}
&Prob\left\{X \geq hH_2(1-3\epsilon-\xi)/e\right\} \\
\geq &Prob\{Z \geq h(1-\xi-\epsilon)\} \qquad Prob\left\{X \geq ZH_2(1-2\epsilon)/e \;\middle|\; Z \geq h(1-\xi-\epsilon)\right\} \\
\geq &1 - e^{-\frac{(hH_2\epsilon)^2}{2e^2}} - e^{-\frac{(h\epsilon)^2}{2e^2}}.
\end{aligned}$$

Recall that the average arrival is bounded as

$$\mathbb{E}\Big\{\sum_{l' \in \hat{\mathcal{I}}_l} A_{l'}(t)\Big\} \leq \frac{H_2}{H_0}(1-5\epsilon-\xi)/e.$$

Let Y denote the total amount of arrival packets in h frames; $Y = \sum_{t=t_{F_0}(1)}^{t_{F_0}(1)+hH_0-1} \sum_{l' \in \hat{\mathcal{I}}_l} A_{l'}(t)$. By using the "upper tail" Hoeffding's Inequality, we obtain

$$Prob\{Y \leq hH_2(1-4\epsilon-\xi)/e\} \geq 1 - e^{-\frac{2h(H_2\epsilon)^2}{H_0|L|^2 A_{max}^2 e^2}},$$

implying that $Prob\{X - Y \geq hH_2\epsilon/e\} \geq 1 - e^{-\frac{(hH_2\epsilon)^2}{2e^2}} - e^{-\frac{(h\epsilon)^2}{2e^2}} - e^{-\frac{2h(H_2\epsilon)^2}{H_0|L|^2 A_{max}^2 e^2}}$. Therefore,

$$Prob\Big\{\sum_{l' \in \hat{\mathcal{I}}_l} Q_{l'}(t_{F_0}(1) + hH_0) \leq V_{2,F_0}(L) - hH_2\epsilon/e\Big\} \geq 1 - \frac{\epsilon'}{|L|}.$$

The two cases above together imply that $\forall l \in L$,

$$Prob\Big\{\sum_{l' \in \hat{\mathcal{I}}_l} Q_{l'}(t_{F_0}(1) + hH_0) \leq V_{2,F_0}(L) - hH_2\epsilon/e\Big\} \geq 1 - \frac{\epsilon'}{|L|}.$$

Hence, considering $V_{2,F_h}(L) = \max_{l \in L} \sum_{l' \in \hat{\mathcal{I}}_l} Q_{l'}(t_{F_0}(1) + hH_0)$, we obtain a bound as

$$Prob\,\{V_{2,F_h}(L) \leq V_{2,F_0}(L) - hH_2\epsilon/e\} \geq 1 - \epsilon'. \quad \square$$

5.6.3 Proof of Proposition 5.7

Let $N_l(t)$ denote the number of transmitting links in $\hat{\mathcal{I}}(l)$ at t; recall that $s_l(t) \in \{0, 1\}$ indicates if l chooses to transmit at time t.

$$\mathbb{E}\{N_l(t)\} = \sum_{l' \in \hat{\mathcal{I}}(l)} \mathbb{E}\{s_{l'}(t)\} \leq \frac{\sum_{l' \in \hat{\mathcal{I}}(l)} V_{1,F}(l') q_k(t)}{\Gamma \sum_{l' \in \hat{\mathcal{I}}(l)} V_{1,F}(l') q_{l'}(t)} \leq \frac{1}{\Gamma}. \quad \square$$

5.6.4 Proof of Lemma 5.12

W.l.o.g., we look at a link $l \in L_i$. Let L' be a maximum independent set in $\mathcal{I}(l)$, where links can choose any power value to use. Let l_{minP} be a link with the smallest transmission power in L', *i.e.*, $\forall l' \in \mathcal{I}(l), P(l_{minP}) \leq P(l')$. It is easy to see that $d(l', l_{minP}) \leq (2\gamma + 1)d_i$. Using this, we can bound the SINR value at $r(l_{minP})$ when only links in L' are transmitting: $\beta \leq SINR(l_{minP}) \leq \dfrac{\sigma^\alpha(2\gamma+1)^\alpha}{(|L'|-1) + N(2\gamma+1)^\alpha d_i^\alpha / P(l_{minP})}$. The above inequality yields $|L'| \leq \frac{\sigma^\alpha(2\gamma+1)^\alpha}{\beta} + 1$. $\square$

Chapter 6

Delay-constrained Throughput Maximization under Graph-based Interference Model

6.1 Preliminaries and Definitions

The network model, traffic model, queueing model and related definitions are described in Chapter 4. For results in Section 6.6, we assume a unit disk model [122], in which each node u has a fixed transmission range (assumed to be 1, w.l.o.g.), and $(u, v) \in L$ if and only if $d(u, v) \leq 1$; two links interfere when one end of a link is within h hops from one end of the other, where h is a constant integer.

6.2 Delay Metric and Problem Definition

The delay metric used in this chapter is average delay, which can be deduced by Little's law in a stable system. We study the average end-to-end delay: (1) for each flow f, as the average time for packets to reach the destination following the flow path; (2) for each session c, as the average time for packets to reach the destination $t(c)$; and (3) for the entire network, as the average over all sessions.

We are now in a position to formally describe the problem we study in this chapter. Given a multi-hop wireless network represented by a graph $G = (\mathcal{V}, L)$ and a set $\mathcal{C}$ of connections with a target delay $\Delta(c)$ (in number of time slots) for each connection c, the goal of the *Delay-Constrained Throughput Maximization* (DCTM) problem is to find a stable rate vector $\lambda()$ that maximizes the total achievable rate $\sum_c \lambda(c)$, while ensuring that the session c per-packet delay is at most $\Delta(c)$. Let OPT denote the maximum total throughput $\sum_c \lambda(c)$ for $\lambda() \in \Lambda^{OPT}$. Let $OPT(\Delta())$ denote the maximal total rate $\sum_c \lambda(c)$ that is feasible under these (delay) constraints. Note that this definition does not restrict us to any scheduling scheme.

As discussed earlier, this problem is computationally hard in general, and our focus is on approximation algorithms. In particular, we develop polynomial-time bi-criteria approximation algorithms; we say an algorithm that computes a rate vector $\lambda()$ gives a (β_1, β_2)-approximation if the total throughput rate guaranteed is at least $\beta_1 OPT(\Delta())$, while the delays are at most $\beta_2 \Delta(c)$ for each session. Note that these are worst case approximation guarantees, which hold for every problem instance.

6.3 Approximation Hardness of DCTM

DCTM is NP-hard: this follows from the fact that even without any delay constraints, the throughput maximization problem is NP-hard [135]. We extend this to show that even simple cases of the DTCM problem are hard even to approximate, if the delay bounds are required to be satisfied; this motivates the need for bi-criteria approximations.

Lemma 6.1. *There is a constant $K' > 0$ such that the* DCTM *problem cannot be approximated within a factor of K' if the interference graph G is a unit disk graph, unless $P = NP$.*

Proof. (Recall that $\chi(G)$ denotes the chromatic number of a graph G, the smallest number of colors needed to color the vertices of G so that no two adjacent vertices share the same color.) Our proof is based on the result of Clark et al. [29], which shows that given a unit disk graph $G = (\mathcal{V}, L)$ (where $\mathcal{V}$ is the nodes set and L is the link set), it is NP-complete to distinguish between the case $\chi(G) = 3$ and $\chi(G) = 4$, where $\chi(G)$ is the chromatic number of G. We reduce this to an instance of DCTM in an interference model based on distance-2 independence model (*i.e.*, two transmissions are simultaneously possible only if the senders are at least distance-2 apart): let $\mathcal{V}'$ be a duplicate node set of $\mathcal{V}$, such that each $v \in \mathcal{V}$ and its counterpart v' in $\mathcal{V}'$ are located very close; let $G'' = (\mathcal{V}'', L'')$ be a graph where $\mathcal{V}'' = \mathcal{V} \cup \mathcal{V}'$, and $L'' = L \cup \{(v, v') : v \in \mathcal{V}\}$. For each node $v \in \mathcal{V}$, we construct n sessions originating at v and ending at each node v'. For each session c, let the target delay $\Delta(c) \triangleq 3$. We assume that the exogenous traffic is at a constant bit-rate for each session c. If $\chi(G) = 3$, observe that a throughput rate of $1/3$ for each connection is possible — within each window of three time steps, all the connections can be scheduled in this interference model, making the total throughput of $n/3$ feasible. On the other hand, if $\chi(G) = 4$, at most $K'n$ of the sessions can be colored using 3 colors (this follows from a simple analysis of the reduction of [29]). Since $\Delta(c) = 3$ for each session, at most $K'n$ sessions can be scheduled within a window of size 3, implying a total throughput of at most $K'n/3$. $\square$

For arbitrary interference graphs (inferred by a general interference model), it can be shown by a reduction from graph coloring that DCTM cannot be approximated within a factor of $O(n^{1-\varepsilon})$, for any $\varepsilon \in (0, 1)$.

6.4 Rate-based Random-access Scheduling RB-Sched

In this chapter, we focus on random-access scheduling, which involves the following process: at each time slot t, each link l stochastically makes channel access attempt with a specific probability $p_l(t)$ (known as *channel access probability*) when $Q_l(t) > 0$; if link l decides to transmit, it will choose a flow f associated with the link with probability $p(l, f)$, defined below. If no collision happens, it will result in successful data transmission, with an ACK packet sent backward at the end of the slot; otherwise, the packet will stay in the queue for the next transmission service. Note that the collision accounts for transmission of both directions on links, due to the definition of interference and the interference sets. In order to simplify our presentation, we ignore the complexity of handling ACKs, which can be done in the same manner as the flows for the sessions, *i.e.*, by reserving a constant fraction of the rate on every link— this only alters our approximation bounds by a constant factor. We focus on *synchronous* random-access scheduling, where all slots are of the same length. In Section 6.7, we also discuss extensions to *asynchronous* random-access scheduling, in which the transmission durations for different links could be different from each other, and from idle slots. The channel access probability for each link l at time t is

$$p_l(t) = 1 - \exp\big(e \sum_{f:\, l \in L(f)} \lambda(f)/(1-\epsilon)\big), \tag{6.1}$$

where $\epsilon \in (0,1)$ denotes a *rate slackness parameter* which can be set before system initiation as a constant. For each transmission on l, $Q_{l,f}$'s packets get serviced with probability

$$p(l,f) = \frac{\lambda(f)}{\sum_{f':\, l \in L(f')} \lambda(f')}.$$

We prove in Section 6.5 that this random-access scheduling scheme is stable if

$$\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_{f:\, l' \in L(f)} \lambda(f) \leq \frac{1-\epsilon}{e},\ \forall l \in L. \tag{6.2}$$

It follows from [28] that the above constraints in Inequality (6.2) with RHS scaled up by $e\mathcal{K}$ are necessary conditions for *any* stable scheduling scheme; note that $\mathcal{K}$ is usually a constant in most interference models. Inequality (6.2) defines a throughput region within a factor of $1/\left(e\mathcal{K}\right)$ of Λ^{OPT}. In other words, this random-access gives an $O(\mathcal{K})$ approximation of the capacity region comparable to that of maximal scheduling, while maintaining a low-complexity distributed manner of operation.

6.4.1 Estimating Mean Traffic Rates

One limitation of the RB-Sched is the requirement of the knowledge of traffic rates. When the traffic rates are not known a' priori, estimation is needed. In many cases the rates can be estimated with a small relative error in n-polynomial time; however, there also exist cases when it requires exponential time in n to obtain estimated rates with high accuracy. Below we provide detailed analysis and results.

Since the arrival process for each connection is all i.i.d., the *mean-ergodic* property holds. That means, the ensemble average of an arrival process can be deduced from a single, sufficiently long sample (realization) of the process. With a similar proof of weak law of large numbers, we show that guaranteed estimation accuracy of the first moment of each arrival process can be obtained simply through time-averaging. We define *estimation error* of mean as the difference between the estimated mean and the real mean, and define *relative estimation error* of mean as the ratio between the estimation error and the real mean. We say a mean rate λ is *polynomially large*, if $\lambda = \Omega(\frac{1}{n^K})$, where $K > 0$ is a constant.

Lemma 6.2. *Within polynomial time in n, the time-average gives an estimated mean rate value with a relative estimation error below an arbitrary constant $\epsilon > 0$, for a traffic flow with a polynomially large mean, w.h.p..*

Proof. Given an i.i.d. arrival process, let $A(t)$ be the amount of arrival packets at time t, and let λ and ν denote the mean and variance respectively. Let $Avg(T) = \frac{1}{T}\sum_{t\leq T} A(t)$. We have $Var(Avg(T)) = \frac{\nu^2}{T}$.

By Chebyshev's Inequality, for any $\epsilon > 0$,

$$Prob\left(|Avg(T) - \lambda| \geq \epsilon\lambda\right) \leq \frac{\nu^2}{\epsilon^2\lambda^2 T}.$$

Therefore, with the choice of $T = \frac{\nu^2}{\epsilon^2\lambda^2}\Theta(n^{2K+2})$, we can ensure that

$$Prob\left(|Avg(T) - \lambda| \geq \epsilon\lambda\right) \leq \frac{1}{n^2}. \qquad \square$$

W.l.o.g., we assume that the total number of arrival traffic flows is $O(n)$, we have the following corollary.

Corollary 6.3. *Within polynomial time in n, the time-averages give estimated mean rate values with a relative estimation error under an arbitrary constant $\epsilon > 0$, for a total of $O(n)$ arrival processes with polynomially large mean rates, w.h.p.*

Although the time spent on mean rate estimation is "short" for those arrival processes with polynomially large means; for arrival processes with polynomially small means, it may require a "long" time for performing time-averaging to ensure a target relative error, for instance, when $\lambda = O(e^{-n})$.

6.5 Delay Upper-bounds for RB-Sched

We now derive end-to-end delay bounds for flows with a given feasible average rate vector $\lambda()$ that satisfies the constraints in Inequality (6.2).

Theorem 6.4. *For a rate vector $\lambda()$ that satisfies Inequality (6.2), the random-access scheduling protocol* RB-Sched *described in Section 6.4 ensures that (i) the system is stable, implying a throughput region of $\frac{\Lambda^{OPT}}{e\mathcal{K}}$; (ii) the average delay for each flow f is $O\left(|L(f)|^2/(\lambda(f))^2\right)$; and (iii) the average network delay is $O\left(\sum_{f\in\mathcal{F}}|L(f)|^2\Big/\left(\sum_{f\in\mathcal{F}}\lambda(f)\min_{f\in\mathcal{F}}\{\lambda(f)\}\right)\right)$.*

We start the proof with the following lower bound on the expected service rate $\mu_{l,f}(t)$ for any flow f and link l; this will be used in all our analysis in the rest of this section. For notational simplicity, we use $x(l,f)=\lambda(f)/(1-\epsilon)$, and $x(l)=\sum_{f\in\mathcal{F}(l)}x(l,f)$. Rewriting Inequality (6.2), we have $\sum_{l'\in\mathcal{I}(l)\cup\{l\}}x(l')\le 1/e$. Then, Equation (6.1) can be rewritten as $p_l(t)=1-e^{-ex(l)}$, and that gives us:

$$\begin{aligned}\mathbb{E}\left\{\mu_{l,f}(t)\right\} &\geq p(l,f)p_l(t)\prod_{l'\in\mathcal{I}(l)}\left(1-p_{l'}(t)\right)\\ &\geq p(l,f)p_l(t)\sum_{l'\in\mathcal{I}(l)}e^{-ex(l')}\\ &\geq p(l,f)\left(1-e^{-ex(l)}\right)e^{ex(l)-1}\\ &=\frac{\left(e^{ex(l)}-1\right)x(l,f)}{ex(l)}\geq x(l,f).\end{aligned}\tag{6.3}$$

The idea for the proof of the above theorem is that due to the properties of random-access scheduling, each flow can be viewed in "isolation" as a tandem system, with lower bounds on the expected service rate of $\mu(l,f)$ that only depend on $x(l,f)$ for each logical queue $Q_{l,f}$, as shown in Equation (6.3). Let the triplet $(Q(),a(),\mu())$ denote a queueing system. From now on till the end of this section, we use R to denote the basic queueing system under the basic scheduling scheme specified in Section 6.4, with the queueing model and the exogenous arrival processes described in Section 6.1. We put R at superscript to denote the quantities of the corresponding system.

We now consider the queues for a specific flow f: $\{Q^R_{i_f,f}\}$, $i\in\{1,2,\ldots,|L(f)|\}$, as a series of tandem queues, and derive delay bounds. Our proof involves two "reductions", which progressively lead to a simpler queueing system with Bernoulli arrival and service processes for the non-source queues, with delays no smaller than those of $\{Q^R_{i_f,f}\}$; additionally, the second queueing system we construct has an increasing sequence of service rates, allowing us to derive end-to-end delay bounds. We start with the following intuitive lemma, followed by the proof.

Lemma 6.5. *Let R' and R'' be two identical queueing systems (with the same initial states and the same set of general i.i.d. arrival processes) but only differ in the service rates: for R', the service rate at time t for the ith link of flow f is $\mu^{R'}_{i_f,f}(t) \in \{0,1\}$, and for R'' it is $\mu^{R''}_{i_f,f}(t) \in \{0,1\}$; $\mathbb{E}\left\{\mu^{R'}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\mu^{R''}_{i_f,f}(t)\right\}$, for each flow f and link i_f, at any time t. Then, for each flow f, the average total queue size in R' is no greater than that in R'',* i.e.,

$$\begin{aligned}&\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\Big\{\sum_{i=1}^{|\mathcal{L}(f)|}Q^{R'}_{i_f,f}(\tau)\Big\}\\ \leq&\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\Big\{\sum_{i=1}^{|\mathcal{L}(f)|}Q^{R''}_{i_f,f}(\tau)\Big\}.\end{aligned}$$

Proof. For each link-flow pair (i_f, f), by taking squares of the following equation $Q_{i_f,f}(t+1) = Q_{i_f,f}(t) - d_{i_f,f}(t) + a_{i_f,f}(t)$, we obtain the following difference of square between the two queueing systems:

$$\begin{aligned}&\big(Q^{R'}_{i_f,f}(t+1)\big)^2 - \big(Q^{R''}_{i_f,f}(t+1)\big)^2\\ =&\big(Q^{R'}_{i_f,f}(t)\big)^2 - \big(Q^{R''}_{i_f,f}(t)\big)^2 + \big(d^{R'}_{i_f,f}(t)\big)^2 - \big(d^{R''}_{i_f,f}(t)\big)^2\\ &+ \big(a^{R'}_{i_f,f}(t)\big)^2 - \big(a^{R''}_{i_f,f}(t)\big)^2 - 2\mu^{R'}_{i_f,f}(t)Q^{R'}_{i_f,f}(t)\\ &+ 2\mu^{R''}_{i_f,f}(t)Q^{R''}_{i_f,f}(t) + 2a^{R'}_{i_f,f}(t)Q^{R'}_{i_f,f}(t)\\ &- 2a^{R''}_{i_f,f}(t)Q^{R''}_{i_f,f}(t) - 2d^{R'}_{i_f,f}(t)a^{R'}_{i_f,f}(t) + 2d^{R''}_{i_f,f}(t)a^{R''}_{i_f,f}(t).\end{aligned}$$

For any pair (f, i_f), $\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big\{a^{R'}_{i_f,f}(\tau)\big\} = \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big\{a^{R''}_{i_f,f}(\tau)\big\} = \lambda(f)$. When $i > 1$, $a_{i_f,f}(t) = d_{i-1_f,f}(t)$, $\big(a_{i_f,f}(t)\big)^2 = a_{i-1_f,f}(t)$, and $\big(d_{i_f,f}(t)\big)^2 = d_{i-1_f,f}(t)$.

Let $\overline{\mu}^{R'}_{i_f,f} \triangleq \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big\{\mu^{R'}_{i_f,f}(\tau)\big\}$, and $\overline{\mu}^{R''}_{i_f,f} \triangleq \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big\{\mu^{R''}_{i_f,f}(\tau)\big\}$.

Taking expectation of, time-averaging and removing equivalent terms on both sides of the above equation yields

$$
\begin{aligned}
0 = &-2\big(\overline{\mu}^{R'}_{i_f,f} - \lambda(f)\big) \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\big\{Q^{R'}_{i_f,f}(\tau)\big\} \\
&+ 2\big(\overline{\mu}^{R'}_{i_f,f} - \lambda(f)\big) \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\big\{Q^{R''}_{i_f,f}(\tau)\big\}.
\end{aligned}
$$

Since $\mathbb{E}\left\{\mu^{R'}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\mu^{R''}_{i_f,f}(t)\right\}$, for each flow f and link i_f, at any time t, we obtain

$$
\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\big\{Q^{R'}_{i_f,f}(\tau)\big\} \leq \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\big\{Q^{R''}_{i_f,f}(\tau)\big\},
$$

which leads to the statement of the lemma. □

(1) **Reduction 1**. We reduce the basic queueing system R for flow f to another tandem system R_1, such that for all $i \in [1, |L(f)|]$, the service rate $\mu^{R_1}_{i_f,f}$ of each queue $Q^{R_1}(i_f, f)$ is a Bernoulli distribution with

$$
\mathbb{E}\left\{\mu^{R_1}_{i_f,f}(t)\right\} = \lambda(f) + \frac{i\varepsilon(f)}{|L(f)|} \leq x^R(i_f, f) \leq \mathbb{E}\left\{\mu^{R}_{i_f,f}(t)\right\},
$$

where $\varepsilon(f) = \epsilon\lambda(f)/(1-\epsilon)$; the exogenous arrival rates remain the same.

Then, Lemma 6.5 implies that $\mathbb{E}\left\{\sum_{i=1}^{|L(f)|} Q^{R_1}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\sum_{i=1}^{|L(f)|} Q^{R}_{i_f,f}(t)\right\}$. Note that whether the reduced system is using wireless medium no longer matters.

(2) **Reduction 2**. Note that the exogenous arrival at the source link of the tandem system $(Q^{R_1}(), a^{R_1}(), \mu^{R_1}())$ is a rather general arrival process, making it nontrivial to use earlier methods directly, *e.g.*, [21,33,57], in bounding the end-to-end delays. We seek to reduce the system R_1 to another queueing system R_2 so that the arrival process for each non-source queue is also Bernoulli. The queueing system R_2 is defined in the following manner: at each time slot t the service rate for each link i_f in L_f on flow f is the same as that for R_1, except that this link tries to access the medium even if the queue $Q^{R_2}_{i_f,f}$ is empty or does not have enough packets to fill in the capacity. In case (i_f, f) gets serviced and $Q^{R_2}_{i_f,f}$ has a backlog smaller than the channel capacity, dummy packets are injected to make full use of the channel capacity during the time slot. These packets will be labeled as packets for flow f. Now that we have unit capacities, each time (i_f, f) accesses the medium, it transmits

one packet to the next queue in line. Therefore, the service process of $Q^{R_2}_{i_f,f}$, and the arrival process at the subsequent queue $Q^{R_2}_{(i+1)_f,f}$ coincide. Note that if the number of retransmission is upper-bounded, the arrival at non-source queues will be smaller. In the system R_1, for any i, $\mu^{R_1}_{i_f,f}$ follows a Bernoulli distribution, which implies that the subsequent arrival is also a Bernoulli process with $a^{R_2}_{(i+1)_f,f}(t) = \mu^{R_2}_{i_f,f}(t) = \mu^{R_1}_{i_f,f}(t)$. This leads to Lemma 6.6:

Lemma 6.6. *For each flow f,*

$$\begin{aligned}
&\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\Big\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R}_{i_f,f}(\tau)\Big\} \\
&\le \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\Big\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R_1}_{i_f,f}(\tau)\Big\} \\
&\le \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\Big\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R_2}_{i_f,f}(\tau)\Big\}.
\end{aligned}$$

(3) **Queueing analysis for** R_2. The fact that the arrival and service processes of each $Q^{R_2}_{i_f,f}$ are subject to Bernoulli distribution, allows us to perform the isolated queueing analysis for each $Q^{R_2}_{i_f,f}$ in isolation. For any link i_f where $i = 1, 2, \ldots, |L(f)|$,

$$\mathbb{E}\left\{\mu^{R_2}_{i_f,f}(t)\right\} = \mathbb{E}\left\{\mu^{R1}_{i_f,f}(t)\right\} = \lambda(f) + \frac{i\varepsilon(f)}{|L(f)|}.$$

Next, we perform Lyapunov drift analysis to derive an upper-bound on the queue size of each $Q^{R_2}_{i_f,f}$. Refer to [43, 106] for the details of this approach. We define the Lyapunov function as

$$\mathbb{L}\left(Q^{R_2}_{i_f,f}(t)\right) \triangleq \left(Q^{R_2}_{i_f,f}(t)\right)^2.$$

The 1-step Lyapunov drift is then defined as:

$$\triangle^{(1)}_{Q}\left(Q^{R_2}_{i_f,f}(t)\right) \triangleq \mathbb{E}\left\{\mathbb{L}\left(Q^{R_2}_{i_f,f}(t+1)\right) - \mathbb{L}\left(Q^{R_2}_{i_f,f}(t)\right)\Big|Q^{R_2}_{i_f,f}(t)\right\}.$$

By referring to Equation 4.3 and Lemma 4.3 of [43], we obtain

$$\begin{aligned}\triangle_Q^{(1)}\left(Q_{i_f,f}^{R_2}(t)\right) \leq & \mathbb{E}\left\{\left(\mu_{i_f,f}^{R_2}(t)\right)^2 + \left(a_{i_f,f}^{R_2}(t)\right)^2 \middle| Q_{i_f,f}^{R_2}(t)\right\} - \\ & \mathbb{E}\left\{2Q_{i_f,f}^{R_2}(t)\left(\mu_{i_f,f}^{R_2}(t) - a_{i_f,f}^{R_2}(t)\right) \middle| Q_{i_f,f}^{R_2}(t)\right\}. \end{aligned} \tag{6.4}$$

Further, $\mathbb{E}\left\{\left(\mu_{i_f,f}^{R_2}(t)\right)^2\right\} = \mathbb{E}\left\{\mu_{i_f,f}^{R_2}(t)\right\} \leq 1/e$, and $\mathbb{E}\left\{\mu_{i_f,f}^{R_2}(t) - a_{i_f,f}^{R_2}(t)\right\} = \varepsilon(f)/|L(f)|$. Additionally, for $i = 1$, we have $\mathbb{E}\left\{\left(a_{1_f,f}^{R_2}(t)\right)^2\right\} \leq A^{(2)}$; when $i > 1$, $\mathbb{E}\left\{\left(a_{i_f,f}^{R_2}(t)\right)^2\right\} = \mathbb{E}\left\{a_{i_f,f}^{R_2}(t)\right\} \leq 1/e$. Inequality (6.4) can be then rewritten as

$$\triangle_Q^{(1)}\left(Q_{i_f,f}^{R_2}(t)\right) \leq \frac{1}{e} + \max\left\{\frac{1}{e}, A^{(2)}\right\} - \frac{2\varepsilon(f)}{|L(f)|} Q_{i_f,f}^{R_2}(t).$$

By taking expectations on both sides of the above inequality and from Theorem 1 in [106] which deduces an inequality from the Lyapunov drift, the mean backlog $\overline{Q}_{i_f,f}^{R_2}$ is

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{Q_{i_f,f}^{R_2}(\tau)\right\} \leq \frac{1+\max\left\{1, eA^{(2)}\right\}}{2e\varepsilon(f)}|L(f)|.$$

Therefore, the sum of mean backlogs for flow f is

$$\sum_{i=1}^{|L(f)|} \overline{Q}_{i_f,f}^{R_2}(t) \leq \frac{1+\max\left\{1, eA^{(2)}\right\}}{2e\varepsilon(f)}|L(f)|^2.$$

Corollary 6.7. *The systems R, R_1 and R_2 are all stable.*

(4) **Average end-to-end delay bound**. By Little's Law, the average delay for flow f's packets is

$$\overline{D}^R(f) = \sum_{i=1}^{|L(f)|} \overline{Q}_{i_f,f}^{R} \Big/ \lambda(f) \leq \sum_{i=1}^{|L(f)|} \overline{Q}_{i_f,f}^{R_2} \Big/ \lambda(f) \leq \frac{1+\max\left\{1, eA^{(2)}\right\}}{2e\varepsilon(f)} \frac{|L(f)|^2}{\lambda(f)}.$$

The average network delay is

$$\overline{D}^R = \sum_{f\in\mathcal{F}} \sum_{i=1}^{|L(f)|} \overline{Q}^R_{i_f,f} \Big/ \sum_{f\in\mathcal{F}} \lambda(f) \le \frac{1+\max\left\{1, eA^{(2)}\right\}}{2e\min\limits_{f\in\mathcal{F}}\{\varepsilon(f)\}} \frac{\sum\limits_{f\in\mathcal{F}} |L(f)|^2}{\sum\limits_{f\in\mathcal{F}} \lambda(f)}.$$

Theorem 6.4 follows by substituting $\epsilon\lambda(f)/(1-\epsilon)$ for $\varepsilon(f)$.

6.6 Multi-commodity Flows with Delay Guarantees

From Theorem 6.4, it follows that the average end-to-end delay bound for flow f is proportional to $|L(f)|^2$, and inversely proportional to $\lambda^2(f)$; the average network end-to-end delay bound is in proportion to $\sum_{f\in\mathcal{F}} |L(f)|^2$, and inversely in proportion to $\sum_{f\in\mathcal{F}} \lambda(f)$ and $\min_{f\in\mathcal{F}} \lambda(f)$. Therefore, in order to find a feasible rate vector $\lambda()$ that minimizes the delay guarantees, we need to construct flows with "high" rate (*i.e.*, to keep $\lambda(f)$ high) and "short" paths (*i.e.*, to make $|L(f)|$ low). In this section, we use terms "path" and "flow" interchangeably.

Given the delay constraint $\Delta(c)$ for each connection c (as defined in Section 6.1), we present a multi-commodity flow framework for choosing a rate vector $\lambda()$ and constructing a set of flows $\mathcal{F}''(c)$ for each session c, with the following properties (recall the definition of $OPT(\Delta())$ in Section 6.1).

Theorem 6.8. *(i) The rate vector $\lambda()$ resulting from our multi-commodity flow framework ensures that:* $\sum_c \lambda(c) = \Omega\left(\frac{\log\log\Delta_m}{\log\Delta_m}\right) OPT\left(\Delta()\right)$. *(ii) For each flow f in the set of flows $\mathcal{F}''(c)$ constructed for each session c if $\lambda(c) > 0$, the rate $\lambda(f) = \Omega\left(\frac{\log\log\Delta_m}{\log\Delta_m}\right)$, and the path has a length of at most $2\Delta(c)$.*

Putting everything together: the total rate is "close" to $OPT\left(\Delta()\right)$, and we have the following delay bounds.

Corollary 6.9. *Using the rate vector $\lambda()$ along with the random-access scheduling scheme described in Section 6.5, we ensure that, for each session c and each flow $f \in \mathcal{F}''(c)$, the average delay is* $O\left(\left(\frac{\log\Delta_m}{\log\log\Delta_m}\Delta(c)\right)^2\right)$.

Our algorithm involves construction of multi-commodity flows with constraints on the paths used; broadly, these constraints bound the sum of the "costs" of the links on the paths, which will be explained later with the LP formulation. Our approach employs an approximation algorithm, which selectively drops some cost-unfavorable sessions and maximizes the rates of the rest of the sessions. Path constrained flows have been studied in a wired setting, *e.g.*, [19,52], but the interference constraints, and the fact that we need both short paths and large flow values make our problem different and difficult. Our algorithm is comprised of the following 3 steps: (1) linear programming formulation, (2) path filtering, and (3) randomized rounding.

(Step 1) **Linear programming formulation**. We use the LP formulation as a basis to compute an upper-bound of $OPT(\Delta())$, and to develop our bi-criteria approximation algorithm. For session c, recall that $\mathcal{F}(c)$ denotes the set of possible paths from $s(c)$ to $t(c)$. We assume a cost function defined on the links; let $cost(l)$ denote the cost of link l. For path f, we define $cost(f) = \sum_{l \in L(f)} cost(l)$ as the cost of path f. The costs can be defined in a fairly general manner. In most of this chapter, the cost of a link l will denote the time (in number of slots) needed for a packet transmission (ignoring queuing and interference delays); therefore, the cost of a path will be proportional to its length. Let $\mathcal{F}(c, L)$ denote the set of paths of cost at most L from $s(c)$ to $t(c)$. As mentioned in Section 4.3.1, we will assume that all link capacities are 1. We start with the following LP formulation (LP) to find a flow vector $y()$ that maximizes $\sum_c y(c)$ subject to Constraints LP-(6.5a) to LP-(6.5e). Here, $y(c)$ denotes the total rate for connection c, and $y(f)$ the rate along path $f \in \mathcal{F}(c)$; $y(l, c) = \sum_{f \in \mathcal{F}(c): l \in L(f)} y(f)$ is the total flow for c along l.

$$\textbf{LP:} \quad \max \quad \sum_c y(c)$$

$$\text{s.t.} \quad \forall c,\ y(c) = \sum_{f \in \mathcal{F}(c)} y(f) \tag{6.5a}$$

$$\forall c,\ \sum_{f \in \mathcal{F}(c)} y(f) cost(f) \leq \Delta(c) y(c) \tag{6.5b}$$

$$\forall l, c,\ y(l, c) = \sum_{f \in \mathcal{F}(c):\ l \in L(f)} y(f) \tag{6.5c}$$

$$\forall l,\ \sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_c y(l', c) \leq \frac{1-\epsilon}{e} \tag{6.5d}$$

$$\forall f,\ y(f) \geq 0 \tag{6.5e}$$

In the above formulation: (1) Constraints LP-(6.5a) and LP-(6.5c) represent path-based flow-conservation constraints. (2) LP-(6.5b) constrains the total path cost, which we use as a lower-bound on the average delay along a flow-path; in our case the cost function is chosen to be path length since end-to-end delay is lower bounded by the number of hops. (3) Congestion constraints in LP-(6.5d) ensure the stability under a random-access scheme; Note that under the h-hop unit disk graph model, since the interference degree $\mathcal{K} = O(h) = O(1)$, $\sum_{l' \in \mathcal{I} \cup \{l\}} \sum_c y(l', c) \leq \mathcal{K} = O(1)$ for any link l.

We write the optimal objective value of (LP) as $OPT^{LP}(\Delta())$. Since LP-(6.5b) serves as relaxed delay constraints and LP-(6.5d) may potentially compromise the optimal value of total flow by at most a constant factor, $OPT^{LP}(\Delta()) = \Omega\big(OPT(\Delta())\big)$.

The above program may have exponentially many constraints because it is formulated using all the flow paths in $\mathcal{F}(c)$, which may include all viable paths in graph G. It is easy to reformulate this as a polynomial sized LP by

(1) replacing Constraints LP-(6.5a) and LP-(6.5c) with for all c,
$\displaystyle\sum_{l \in L_{out}(s(c))} y(l, c) = y(c)$ and $\displaystyle\sum_{l \in L_{in}(t(c))} y(l, c) = y(c)$, and flow-conservation constraints at all other nodes;

(2) replacing LP-(6.5b) with for all c,
$\sum_{l \in L} y(l, c) cost(l) \leq \Delta(c) y(c)$; and

(3) replacing LP-(6.5e) with for all l, c, $y(l, c) \geq 0$.

Let $y^*()$ denote the optimum fractional solution to the above LP; $OPT^{LP}(\Delta()) = \sum_c y^*(c)$. Following standard techniques, *e.g.*, [6], this flow can be decomposed into path flows $y^*(f)$ in polynomial time, with a polynomial number of paths that have positive flow. Let $\mathcal{F}^* = \{f : y^*(f) > 0\}$ be the set of flows with positive flow.

(Step 2) **Filtering**. The LP solution might result in some flows on long paths. For any session c, let $\mathcal{F}^*(c, 2\Delta(c)) = \{f \in \mathcal{F}^*(c) : |\mathcal{L}(f)| \leq 2\Delta(c)\}$ be the set of flows in $\mathcal{F}^*(c)$ with path lengths bounded by $2\Delta(c)$. We transform $y^*()$ into another fractional solution $y'()$ in the following manner, to avoid long paths:

$$\forall f,\ y'(f) = \begin{cases} y^*(f), & \text{if } f \in \mathcal{F}^*(c, 2\Delta(c)); \\ 0, & \text{otherwise} \end{cases}$$

It follows by a simple averaging argument, that

$$\forall c,\ y'(c) = \sum_{f \in \mathcal{F}^*(c, 2\Delta(c))} y'(f) \geq y^*(c)/2. \tag{6.6}$$

Let $\mathcal{F}' = \{f \in \mathcal{F} : y'(f) > 0\}$ be the set of flows with positive flow; for each $f \in \mathcal{F}'$, we have $cost(f) \leq 2\Delta(c)$.

(Step 3) **Randomized rounding**. In this step, we round the filtered solution to an integral solution to obtain Lemma 6.10.

Lemma 6.10. *After the randomized rounding step, we obtain a set $\mathcal{F}'' \subseteq \mathcal{F}^*$ of paths with positive rates, and a rate vector $\lambda()$, such that (1) for each $f \in \mathcal{F}''$, $\lambda(f) = \Omega\left(\log\log\Delta_m / \log\Delta_m\right)$; (2) $\sum_c \lambda(c)$ is $\Omega\left(\log\log\Delta_m / \log\Delta_m\right) OPT^{LP}(\Delta())$; and (3) the chosen paths incur "low" congestion; more precisely, for each link l, we have $\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_c \lambda(l', c) \leq \frac{1-\epsilon}{e}$, where $\lambda(l', c)$ is the rate on all paths $f \in \mathcal{F}''(c)$ such that $l' \in L(f)$.*

We first describe the sub-steps of the rounding stage and then discuss the proof of Lemma 6.10.

(1) **Pre-processing:** we partition paths into groups, formulate a minimax integer program (MIP) that minimizes maximum congestion and that chooses one path in each group with "large" flow rate, and formulate a relaxation of the minimax integer program with refined paths.
(2) **Randomized rounding:** we employ the techniques based on [131] to derive an approximate solution to the relaxed minimax integer program.
(3) **Post-processing:** we scale down the flow rates by a "reasonably small" factor, such that the congestion constraints LP-(6.5d) can be satisfied.

The details are provided below.

(Step 3.1) **Pre-processing: Bin-packing**. Let l_{max} be such that $\mathcal{I}(l_{max})$ has the maximum congestion under with $y'()$ and $\mathcal{F}'$, *i.e.*,

$$l_{max} \triangleq \arg\max_{l \in L} \sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_{f \in \mathcal{F}' : l' \in L(f)} y'(f).$$

Define $\mathcal{F}'_{max}$ as the set of all paths that touch $\mathcal{I}(l_{max})$:

$$\mathcal{F}'_{max} \triangleq \bigcup_{l \in \mathcal{I}(l_{max}) \cup \{l_{max}\}} \{f : l \in L(f)\}.$$

We partition $\mathcal{F}'$ into a sequence of groups $\mathcal{F}'_1, \mathcal{F}'_2, \ldots$ of paths, where the first group $\mathcal{F}'_1 = \mathcal{F}'_{max}$, and $\mathcal{F}'_2, \mathcal{F}'_3, \ldots$ are constructed in an arbitrary manner, such that for each i (for all but possibly one group),

$$\sum_{f \in \mathcal{F}'_{max}} y'(f) \leq \sum_{f \in \mathcal{F}'_i} y'(f) < 2 \sum_{f \in \mathcal{F}'_{max}} y'(f).$$

Let k denote the total number of such path groups. Due to our construction, if there is a group $\mathcal{F}'_i$ with $\sum_{f \in \mathcal{F}'_i} y'(c) < \sum_{f \in \mathcal{F}'_{max}} y'(f)$, we have $i = k$. It is easy to see that $k = \Theta(OPT^{LP})$, since $\sum_{f \in \mathcal{F}'_{max}} y'(f) = \Theta(1)$ according to (LP).

(Step 3.2) **Pre-processing: MIP formulation**.

$$\begin{aligned}
\textbf{MIP:} \quad & \min \quad w \\
& \text{s.t.} \quad \sum_{f \in \mathcal{F}'_i} z(f) = 1, \ \forall i = 1, \ldots, k && \text{(6.7a)} \\
& \qquad \sum_{l' \in \mathcal{I}(l) \cup \{l\}} \ \sum_{f \in \mathcal{F}' : l' \in L(f)} z(f) \leq w, \ \forall l && \text{(6.7b)} \\
& \qquad z(f) \in \{0, 1\}, \ \forall f \in \mathcal{F}' && \text{(6.7c)}
\end{aligned}$$

(MIP) above formulates a minimax integer program that minimizes maximum congestion among all the interference sets. Constraints MIP-(6.7a) and MIP-(6.7c) let us choose one flow path from each set $\mathcal{F}'_i$, and assign flow rate of 1 to the paths chosen. Since $\sum_{f \in \mathcal{F}'_{max}} y'(f) = \Theta(1)$, the vector y', after suitable scaling is a feasible solution for the linear relaxation of (MIP); further, it will turn out that the objective value of an optimum fractional solution to (MIP) is $O(1)$. MIP-(6.7b) can be rewritten in an aggregate manner as $\mathbf{B}\vec{z} \leq \vec{w}$, where $\mathbf{B}$ is a $|L| \times |\mathcal{F}'|$ matrix. Note that $|\mathcal{F}'|$ is polynomial in $|L|$. Intuitively, this integer program is hard to solve exactly, because matrix $\mathbf{B}$ can be dense and irregular, as a result of the facts that an interference set can be as large as L and that both of the set of flow paths in an optimal solution and the graph topology are non-controllable. In light of this, we perform the following path refinement (Step 3.3), constraint relaxation (Step 3.4) and MIP reformulation (Step 3.5) to approximately solve (MIP). It is in these steps that we require an h-hop unit disk graph (UDG) model as described in Section 6.1.

(Step 3.3) **Pre-processing: Path refinement**. For each link $l = (u, v)$, if there is a path $f \in \mathcal{F}'$ that uses more than a constant number, K_0, of links in $\mathcal{I}(l)$, we "short-cut" f into f' that uses at most K_0 such links, and does not violate any of the constraints of (LP). This is illustrated in Figure 6.1.

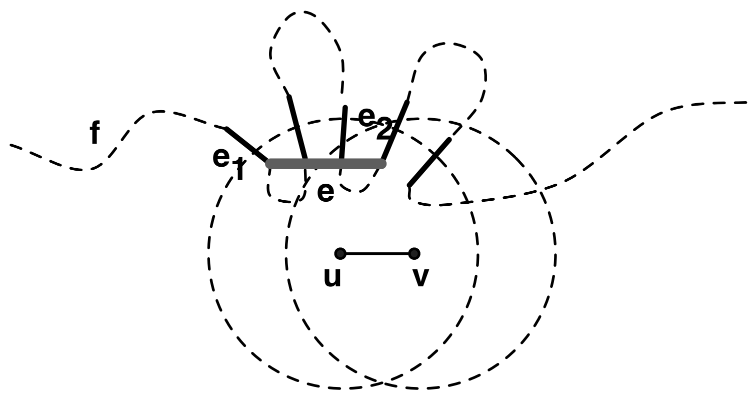


Figure 6.1: Path refinement operation: Consider path f (shown by dashed curved line) and link $l = (u, v)$, in a UDG model. Path f revisits $\mathcal{I}(l)$ multiple times, and the segment of f from between the endpoints of edges e_1 and e_2 can be replaced by edge e (shown in light gray and bold) to get a path f' which is shorter; sending the same flow on f' (instead of f) is still feasible. We then continue to use $\mathcal{F}'$ to denote the refined set of paths without ambiguity.

(Step 3.4) **Pre-processing: Relaxation of congestion constraints**. Recall that we have rewritten MIP-(6.7b) into $\mathbf{B}\vec{z} \leq \vec{w}$. Let $\eta(\mathbf{B})$ denote the maximum number of congestion constraints (in MIP-(6.7b)) in which a path in $\mathcal{F}'$ is simultaneously involved; the definition is based on the fact that a path f appears in every congestion constraint corresponding to the links on f and those interfering with links on f. It is crucial that $\eta(\mathbf{B})$ be "small", so that the rounding scheme in later steps produces a "good" approximate ratio. However, $\eta(\mathbf{B})$ may become $\Omega\left(\Delta_m \max_l |\mathcal{I}(l)|\right)$ if we use the original set of congestion constraints in LP-(6.5d), which is the case for general interference model. To control this, in a UDG model, we now coarsen the formulation in the following manner:

(1) We partition the plane into $\frac{1}{8} \times \frac{1}{8}$ grid cells. A cell is said to be *non-empty* if there is at least one node in it. Let $\mathcal{B}$ denote the set of non-empty cells and let b denote a cell in $\mathcal{B}$. We say a link $l \in b$ if and only if l is within a constant distance of any point in cell b.

Since the path refinement do not increase path lengths, all the path lengths are bounded by $2\Delta_m$. The number of cells that a path $f \in \mathcal{F}'$ goes through is hence $O(\Delta_m)$.

(2) MIP-(6.7b) implies the following:

$$\forall b \in \mathcal{B}, \sum_{l \in b} \sum_{f \in \mathcal{F}': l \in L(f)} z(f) \leq w. \tag{6.8}$$

We scale the coefficients of any rate variable $z(f)$ in Inequality (6.8) down by a constant factor of K_0 (which upper-bounds the number of links of a path that lie in the same interference set (as a result of Step 3.3)), such that the coefficients in the inequality above falls in $[0, 1]$. That renders a relaxed set of congestion constraints as

$$\forall b \in \mathcal{B}, \frac{1}{K_0} \sum_{l \in b} \sum_{f \in \mathcal{F}': l \in L(f)} z(f) \leq w.$$

This can be rewritten in an aggregate manner as $\mathbf{B}'\vec{z} \leq \vec{w}$, where $\mathbf{B}'$ is a $[0, 1]^{|\mathcal{B}| \times |\mathcal{F}'|}$ matrix (with $|\mathcal{B}| \leq |\mathcal{V}|$ as the number of non-empty cells). After the relaxation, $\eta(\mathbf{B}') = O(\Delta_m)$.

(Step 3.5) **Pre-processing: MIP reformulation**. We reformulate (MIP) into (MIP-1) as below:

$$\begin{aligned}
\textbf{MIP-1:} \quad & \min \quad w \\
& \text{s.t.} \quad \sum_{f \in \mathcal{F}'_i} x(f) = 1, \ \forall i = 1, \ldots, k && \text{(6.9a)} \\
& \qquad \mathbf{B}'\vec{x} \leq \vec{w} && \text{(6.9b)} \\
& \qquad x(f) \in \{0, 1\}, \ \forall f \in \mathcal{F}' && \text{(6.9c)}
\end{aligned}$$

Since any interference set in an h-hop UDG model touches at most $O(h^2)$ cells, any solution to the (MIP-1) with an objective value w is a solution to (MIP) and produces an objective value of (MIP) which is $O(h^2 K_0)w = O(w)$, where h and K_0 are constant.

(Step 3.6) **Rounding process**. Applying the rounding algorithm of [131] to solving (MIP-1) yields a rate vector $x'()$, such that a path f_i with rate $x'(f_i) = 1$ is chosen from $\mathcal{F}'$ for each group $\mathcal{F}'_i$ (except group $\mathcal{F}'_k$, in case $\sum_{f \in \mathcal{F}'_k} y'(f) < \sum_{f \in \mathcal{F}'_1} y'(f)$); and for the rest of the paths not chosen, zero flows are assigned.

Lemma 6.11. *Let $\mathcal{F}'' = \{f : x'(f) > 0\}$ denote the set of selected paths with positive rates; $\mathcal{F}'' \subseteq \mathcal{F}'$. The rounding process ensures that (1) for each $f \in \mathcal{F}''$, $x'(f) = 1$; (2) for each connection c, $x'(c) = \sum_{f \in \mathcal{F}''(c)} x'(f)$. (3) $|\mathcal{F}''| = \sum_c x'(c) = \Theta(\sum_c y^*(c)) = \Theta\big(OPT^{LP}(\Delta())\big)$; and (4) for each link l, we have $\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_c x'(l', c) \leq K_1 \log \Delta_m / \log\log \Delta_m$, where $x'(l', c)$ is the flow rate on path $f \in \mathcal{F}''(c)$ such that $l' \in L(f)$, and K_1 is a constant.*

In [131], randomized rounding techniques are proposed for solving a minimax integer program (MIP-2) as shown below.

$$\begin{aligned}
\textbf{MIP-2:} \quad & \min \quad w \\
& \text{s.t.} \quad \sum_{j \in \mathcal{X}_i} x_{i,j} = 1, \ \forall i \in \mathcal{X} && \text{(6.10a)}\\
& \qquad \mathbf{A}\vec{x} \leq \vec{w} && \text{(6.10b)}\\
& \qquad x_{i,j} \in \{0, 1\}, \ \forall i \in \mathcal{X}, \ \forall j \in \mathcal{X}_i && \text{(6.10c)}
\end{aligned}$$

(MIP-2) seeks to minimize maximum components in $\mathbf{A}\vec{x}$ while satisfying the equality constraints MIP-2-(6.10a) and the integrity constraints MIP-2-(6.10c). $\mathcal{X}$ is a set of distinct integers, and $\mathcal{X}_i$ is a set of distinct integers associated with integer $i \in \mathcal{X}$. The total number of $x_{i,j}$ variables is denoted by $N = \sum_i |\mathcal{X}_i|$. $\vec{x}$ denotes the N-dimensional vector of variables $x_{i,j}$. $\mathbf{A} \in [0, 1]^{M \times N}$ is an $M \times N$ matrix, and $\vec{w}$ is an M-dimensional vector with variable w in each component.

Lemma 6.12 (Lemma 2.4(a) in [131])**.** *Given independent r.v.s $X_1, \ldots, X_n \in [0, 1]$, let $X = \sum_{i=1} X_i$ and $\mu = \mathbb{E}\{X\}$. Then, $\forall \mu > 0$, $\forall p \in (0, 1)$, $\exists \delta = H(\mu, p) > 0$, such that $\lceil \mu\delta \rceil (e^{\delta}/(1 + \delta)^{1+\delta})^{\mu} \leq p$, and such that*

$$H(\mu, p) = \begin{cases} \Theta(\frac{\log(p^{-1})}{\mu \log(\mu^{-1} \log(p^{-1}))}) \ \textit{if } \mu \leq \frac{1}{2}\log(p^{-1}); \\ \Theta(\sqrt{\mu^{-1} \log(\mu + p^{-1})} \ \textit{otherwise.} \end{cases}$$

For any matrix $\mathbf{M}$, let $\gamma(\mathbf{M})$ denote the maximum number of non-zero entries in any column of $\mathbf{M}$.

Theorem 6.13 (Theorem 2.5 in [131])**.** *Let w^*_{LP} denote the optimum value of the LP relaxation of (MIP-2). Then there exists an integral solution of value at most $w^*_{LP} + O(1) + O(\min\{w^*_{LP}, M\} H(\min\{w^*_{LP}, M\}, 1/\gamma(\mathbf{A})))$ for (MIP-2).*

proof of Lemma 6.11. We observe that (MIP-1) is an instance of (MIP-2). Recall that $\gamma(\mathbf{A})$ denotes the maximum number of non-zero entries in any column of $\mathbf{A}$; note that $\gamma(\mathbf{B}') = \eta(\mathbf{B}') = O(\Delta_m)$, and we will refer to this later. Since w^*_{LP} is the optimum value of the LP relaxation of (MIP-2), w^*_{LP} is a lowerbound on the optimum value of (MIP-2). According to Lemma 6.12 and Theorem 6.13, if (MIP-2) satisfies the three conditions —

(1) $w^*_{LP} = \Omega(1)$,
(2) $\gamma(\mathbf{A}) > \max\left\{1, 2^{2w^*_{LP}}\right\}$, and
(3) $M > w^*_{LP}$

— then the rounding scheme yields an integral solution to (MIP-2) with an objective value of $O\left(\frac{\log \gamma(\mathbf{A})}{\log\log \gamma(\mathbf{A})}\right)$.

Now we argue that (MIP-1) satisfies the three conditions. With a little abuse of the notation, we use w^*_{LP}, M for the same meanings in the context of (MIP-1). Let rate vector $x^*_{LP}()$ denote an optimal solution to the LP relaxation of (MIP-1).

(1) Because the set $\mathcal{F}'_1$ have been chosen in a way that it only contains the paths that touch the most congested interference set, there exists a cell b that covers part of the most congested interference set such that

$$\sum_{l \in b} \sum_{f \in \mathcal{F}': l \in L(f)} x^*_{LP}(f)/K_0 \geq \sum_{f \in \mathcal{F}'_1} x^*_{LP}(f)/O(h^2 K_0) \geq 1/O(h^2 K_0).$$

That implies $w^*_{LP} = \Omega(1)$.

(2) W.l.o.g., we assume at least one connection requires more than one hop. We construct a feasible solution x'_{LP} for the LP relaxation of (MIP-1) and show that the corresponding objective value w'_{LP} is at most 1. For each i and each $f \in \mathcal{F}'_i$, we assign $x'_{LP}(f) = y'(f)/\sum_{f' \in \mathcal{F}'_i} y'(f')$, such that constraints MIP-1-(6.9a) are satisfied. On the one hand, any path of two or more hops touches at least 6 cells; that implies $\gamma(\mathbf{B}') \geq 6$. On the other hand, because a path has at most K_0 links in any interference set, $\sum_{l \in b} \sum_{f \in \mathcal{F}': l \in L(f)} y'(f) \leq K_0 \sum_{f \in \mathcal{F}'_1} y'(f)$; and since $\forall f \in \mathcal{F}', x'_{LP}(f) \leq y'(f)/\sum_{f' \in \mathcal{F}'_1} y'(f')$, the congestion incurred by $x'_{LP}()$ in each cell b is

$$\sum_{l \in b} \sum_{f \in \mathcal{F}': l \in L(f)} x'_{LP}(f) \leq K_0.$$

The objective value under the rate vector $x_{LP}(f)$ is thus $w'_{LP} \leq K_0/K_0 = 1$, which implies $w^*_{LP} \leq 1$. Therefore, $\gamma(\mathbf{B}') > \max\left\{1, 2^{2w^*_{LP}}\right\}$.

(3) $M = |\mathcal{B}| \geq \gamma(\mathbf{B}') > w^*_{LP}$.

Therefore, by applying the rounding algorithm, we obtain a set $\mathcal{F}'' = \{f_i\}$ of paths with positive rates and a rate vector $x'()$ that satisfies

$$x'(f_i) = \sum_{f \in \mathcal{F}'_i} x'(f) = 1 = \Theta\big(\sum_{f \in \mathcal{F}'_i} y'(f)\big), \forall i.$$

That can be translated to

$$\begin{aligned}|\mathcal{F}''| &= \sum_c x'(c) = \sum_i \sum_{f \in \mathcal{F}'_i} z'(f) \\ &= \Theta\big(\sum_c y'(c)\big) = \Theta\big(\sum_c y^*(c)\big) = \Theta\big(OPT^{LP}(\Delta())\big).\end{aligned}$$

Since $\gamma(\mathbf{B}') = O\left(\Delta_m\right)$, the congestion at each cell is at most

$$O(\log \gamma(\mathbf{B}')/\log\log \gamma(\mathbf{B}')) = O(\log \Delta_m/\log\log \Delta_m),$$

which implies that the congestion at each interference set is $O(h^2 \log \Delta_m/\log\log \Delta_m) = O(\log \Delta_m/\log\log \Delta_m)$. Hence, Lemma 6.11 holds. □

(Step 3.7) **Post-processing: Scaling and choosing flow vector**. We choose a rate vector $\lambda()$ as $\lambda(f) = K_2 \frac{\log\log \Delta_m}{\log \Delta_m} x'(f)$, $\forall f \in \mathcal{F}''$, where K_2 is a constant, such that the congestion constraints in LP-(6.5d) are satisfied. Note that for some connections, multiple flows might be chosen (whereas for some connections, none would be chosen); for each ("original") connection c, define $\lambda(c) = \sum_{f \in \mathcal{F}''(c)} \lambda(f)$ as the total flow of c. $\sum_c \lambda(c) = \frac{\log\log \Delta_m}{\log \Delta_m} \Omega\big(OPT^{LP}(\Delta())\big)$. As discussed before, $OPT^{LP}(\Delta()) = \Omega\big(OPT(\Delta())\big)$. Now Lemma 6.10 follows, and our algorithm ends.

Combining the delay analysis in Section 6.5, gives us Theorem 6.8 and Corollary 6.9 — the bi-criteria approximation — for DCTM problem.

6.7 Asynchronous Random-access Scheduling

The asynchronous random-access scheduling scheme is based on mechanisms of the 802.11 protocol. The smallest unit of time has a length of T_{id}. Let l be an arbitrary link in G. The interference set $\mathcal{I}(l)$ is partitioned into two sets: (1) $\mathcal{I}_{exp}(l)$ that consists of all links that can sense link l's transmission and (2) $\mathcal{I}_{hid}(l) = \mathcal{I}(l) \backslash \mathcal{I}_{exp}(l)$, where "exp" means exposed and "hid" means hidden. When link l senses no interfering signals, it attempts to transmit with probability $p_l(t)$. Once link l gets the channel without collision, it occupies the channel for $T_{tx}(l)$ time until its backlog is empty.

The work in [26] gives the following results. Let γ denote the maximum ratio between $T_{tx}(l)$ of any link l and $T_{tx}(l')$ of any hidden interfering link l' of l, *i.e.*, $\gamma = \max_l \max_{l' \in \mathcal{I}_{hid}(l)} \frac{T_{tx}(l)}{T_{tx}(l')}$. For a system under asynchronous random-access scheduling, we set the channel access probability of each link l:

$$p_l(t) = 1 - e^{-e \sum_{f: l \in L(f)} (\lambda(f)/(1-\epsilon)) T_{id}/T_{tx}(l)},$$

where $0 < \epsilon < 1$; then we can achieve a throughput region of $\frac{1}{e(\gamma+1)\mathcal{K}} \Lambda^{OPT}$, when the arrival vector $\lambda()$ satisfies

$$\sum_{l' \in \mathcal{I}_{hid}(l)} \sum_{f \in \mathcal{F}(l')} \lambda(f) \frac{T_{tx}(l') + T_{tx}(l) - T_{id}}{T_{tx}(l')} + \sum_{l' \in I_{exp}(l)} \sum_{f: l' \in L(f)} \lambda(f) \leq \frac{1-\epsilon}{e(\gamma+1)}, \ \forall l \in L.$$

The expected service rate for queue $Q_{l,f}$ on l at any time t is lower-bounded as $\mathbb{E}\{\mu_{l,f}(t)\} \geq \lambda(f)/(1-\epsilon)$. Details can be found in [26]. By using the queueing reduction technique in Section 6.5, we obtain a similar delay bounds as in Theorem 6.4. Plugging in the stability constraints to (LP) in Section 6.6, we obtain the throughput-delay guarantees as in Theorem 6.14.

Theorem 6.14. *(i) The rate vector $\lambda()$ resulted by our multi-commodity flow framework ensures that: $\sum_c \lambda(c) = \Omega\left(\frac{\log\log\Delta_m}{\gamma\log\Delta_m}\right) OPT\left(\Delta()\right)$. (ii) For each flow f in the set of flows $\mathcal{F}''(c)$ constructed for each session c if $\lambda(c) > 0$, the rate $\lambda(f) = \Omega\left(\frac{\log\log\Delta_m}{\gamma\log\Delta_m}\right)$, and the path has length at most $2\Delta(c)$. (iii) Using the random-access scheduling protocol, we ensure that, for each session c and each flow $f \in \mathcal{F}''(c)$, the average delay is $O\big((\frac{\gamma\log\Delta_m}{\log\log\Delta_m}\Delta(c))^2\big)$.*

6.8 Quantifying the Delays from Adaptive Channel Switching

While adaptive channel-switching capabilities in recent MC-MR and cognitive radio devices have led to throughput improvements [78, 91], the switching delays might not be negligible, and this affects the end-to-end delays. We show how our formulation can incorporate these delays to study multi-channel systems.

Here, we discuss a single radio interface per node; this can be easily extended to the case of multiple interfaces. Let Ψ denote the set of channels available in the system; let ψ, ψ' be two arbitrary channels in Ψ. If l and l' are incoming and outgoing links of a node respectively, let the delay in switching from channel ψ to ψ' be denoted by $d(\psi, \psi')$. Our formulation in Section 6.6 is based on link delays, whereas switching delays are not captured because they are associated with nodes. The difficulty of applying the LP formulation lies in adapting constraints LP-(6.5c) and LP-(6.5d) to multi-channel model. We tackle this by performing a graph transformation on the network graph G to a new graph G' by the following three steps: (1) We *split* each link in G into $|\Psi|$ links, each associated with a unique channel; (2) for each node $v \in G$, we *split* it into $\big(|L_{in}(v)| + |L_{out}(v)|\big)|\Psi|$ nodes, each of which is incident with only one incoming or outgoing link. (3) each node incident with an incoming link is connected to each node incident with an outgoing link, by an intermediate link associated with a switch delay.

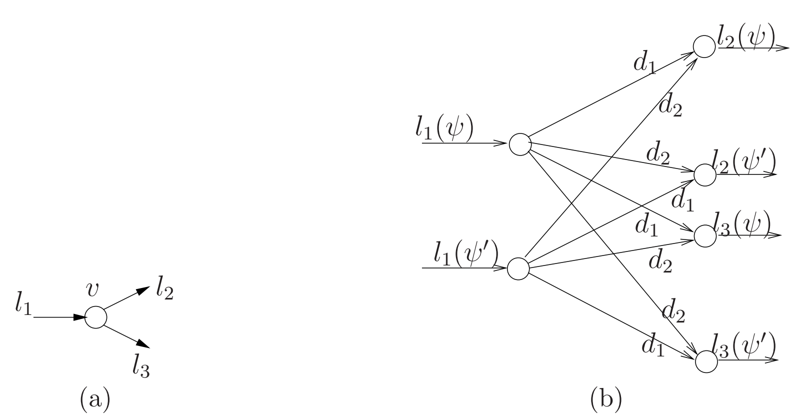


Figure 6.2: (a) Node v with incoming link l_1, outgoing links l_2, l_3, and channels ψ, ψ'. (b) The reduction after node and link splitting with addition of switching link with delays d_1, d_2.

Let $l(\psi)$ denote the link associated with channel ψ in G' emerged from link l in G, Let L' denote the set of links in G'. Figure 6.2 shows an example of transforming the original network graph in Figure 6.2a to the graph in Figure 6.2b, where switch delays are $d_1 = d(\psi, \psi) = d(\psi', \psi')$ and $d_2 = d(\psi, \psi') = d(\psi', \psi)$. For node v, let $L'_{(1)}(v)$ denote the new sets of links emerging from Step (1) above, which corresponds to the incoming and outgoing links in Figure 6.2a; and let $L'_{(3)}(v)$ denote the sets of new links emerging from Step (3) above, which corresponds to the set of all the complete bipartite links in the middle connecting the incoming and outgoing links in Figure 6.2b.

For link $l \in G$, let $Pri(l)$ denote the primary interference set which includes all links in G sharing an end with link l. After graph transformation, for link $l(\psi) \in G'$, let $\lambda(l(\psi)) \triangleq \sum_{c \in \mathcal{C}} \lambda\left(l(\psi), c\right)$. The stability condition [56] is

$$\lambda\left(l(\psi)\right) + \sum_{\psi' \in \Psi \setminus \{\psi\}} \lambda\left(l(\psi')\right) + \sum_{\psi' \in \Psi} \sum_{l' \in Pri(l)} \lambda\left(l'(\psi')\right) + \sum_{l' \in \mathcal{I}(l) \setminus Pri(l)} \lambda\left(l'(\psi)\right) \leq \frac{1-\epsilon}{e}, \ \forall l, \forall \psi.$$

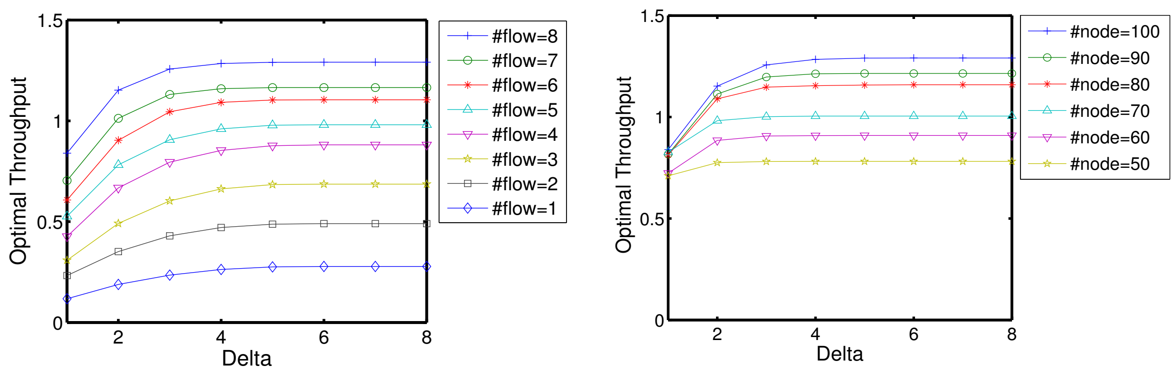
In the above inequality, note that l and l' denote links from G; the link-channel pair $l(\psi)$ denotes a link from G'. Additionally, we construct interference constraints on the intermediate switching links in G', depending on specific switching conditions. For example, when we are restricted to using only one channel at a time, we can apply the following as the interference constraints for intermediate switching links:

$$\sum_{l \in L'_{(1)}(v) \cup L'_{(2)}(v)} \lambda(l) \leq 1, \forall v.$$

Then after modifying the flow conservation conditions and using switching delay as the link cost for any intermediate switching link, we are able to adapt the LP in Section 6.6 to finding a multi-commodity flow vector for the multi-channel model. By using the distributed random-access scheduling scheme of [56], and setting $p(l, f)$ as in Section 6.1, we can obtain results in the same form as Theorem 6.8.

6.9 Simulation Results

We study the performance of our algorithms empirically with simulation. First, for single-channel models, we show how the optimal network throughput depends on varying uniform



(a) Throughput-delay trend with varying #flows on 100-node topo

(b) Throughput-delay trend with 8 flows and varying network sizes

Figure 6.3: tradeoffs among OPT throughput, delay, number of flows, network size

target delay, number of sessions and network size. Next, for a multi-channel network, we study the variation in the optimal throughput as the number of channels and Δ values vary. Experiments are carried out both on random unit disk graph topologies and grid network topologies, with both primary interference and two-hop interference models. LP's are solved with SCIP [2] and SoPlex [139] bundle.

6.9.1 Single-channel Networks

We generated random unit disk graphs with varying sizes, and varied the number of random connections for a network topology. For each choice of network size, number of connections and Δ value, we perform 500 iterations of random topology and connection generation, plus LP formulation.

Figure 6.3 shows the throughput-delay tradeoffs for different number of flows and different network size under the same interference model. Figure 6.3a features a fixed network size of 100, and Figure 6.3b features a fixed number of flows equal to 8. Intuitively, as Δ values increase, thereby loosening the delay constraint, the optimal throughput will rise; as the number of random connections goes up, throughput increases, since the optimization process gets more exploration space. The saturation of the curves happen where the interference plays a major role through the congestion constraints in the LP.

Figure 6.4 shows the impact of different levels of interference severance by using primary

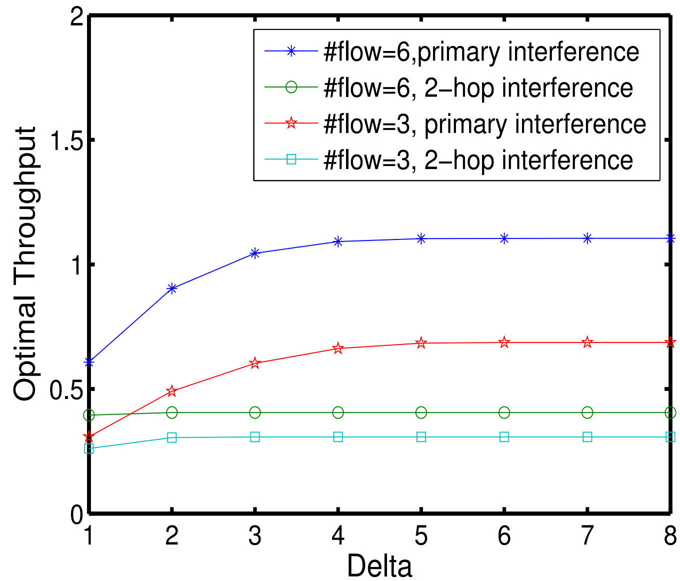


Figure 6.4: Impact of interferences: primary interference and two-hop interference

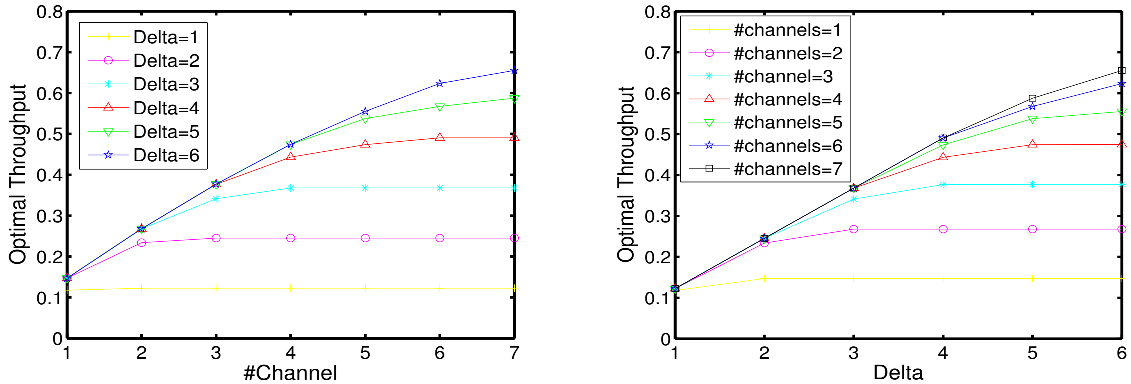
and 2-hop interference models, both on 100-node topologies. The increased interference severance (due to added flows and extended interference) causes the network throughput to reach a saturation point faster.

6.9.2 Multi-channel Networks

Figure 6.5a and 6.5b show the optimal throughput calculated by solving the LP's for grid topologies with 2-hop interference model on grid topologies. As expected, the total throughput increases as additional channels are equipped and delay bounds are relaxed. Saturation points are observed in both plots. Addition of channel resources alleviates the interference, thus yielding a slower saturation process. Also, loosening the delay bound produces similar effects, and the addition of channels make the optimization process to exploit more under the delay bounds.

6.9.3 Distribution of Individual Flow Packet Delay

Figure 6.6 shows the cumulative distribution of individual flow packet delays. We conducted this set of experiments on 50-node network topologies with 5 flows that have i.i.d. arrivals



(a) Optimal throughput v.s. # channels, with fixed delay bounds.

(b) Optimal throughput v.s. delay bound, with fixed # channels.

Figure 6.5: tradeoffs among OPT throughput, delay, number of channels

and random paths, operating with a single channel. We run the simulation for 500 iterations. In Figure 6.6, we show the results for flows with path lengths (*i.e.*, number of hops) of 24, 16, 8, 7, 7 respectively. We notice that over 90% of the packets of each flow experience session delays linear in the number of hops. Further, the CDF curves are almost centro-symmetric within the 0.1% and 99.9% range, and the total span of the packet delays of each flow is small and is densely around the line in which over 90% of the packets of a flow attribute their delay distribution to. That means that with a high probability, all packet delays are subject to a reasonably small upper-bound. Moreover, the packet delay CDF for flows with the same path length tend to be the same, as can be seen from the two flows of lengths 7; we also have the same observations for flows of other lengths on other topologies (omitted here).

In Figure 6.7, we show the average ratios of packet delay to path length for the flows with the same mean arrival rate. These CDF curves, which lie close to each other, share a small span of values and almost the same mean value, meaning that on a topology the flows that have the same arrival rate tend to share the same ratio of delay to path length under our random-access scheduling scheme. We also observe that the average individual packet delays tend to have a much lower growth rate than our analytical quadratic delay bound (which may be too conservative); that suggests that our random-access scheduling scheme may yield close-to-linear (or close-to-optimal) delay performance in practice.

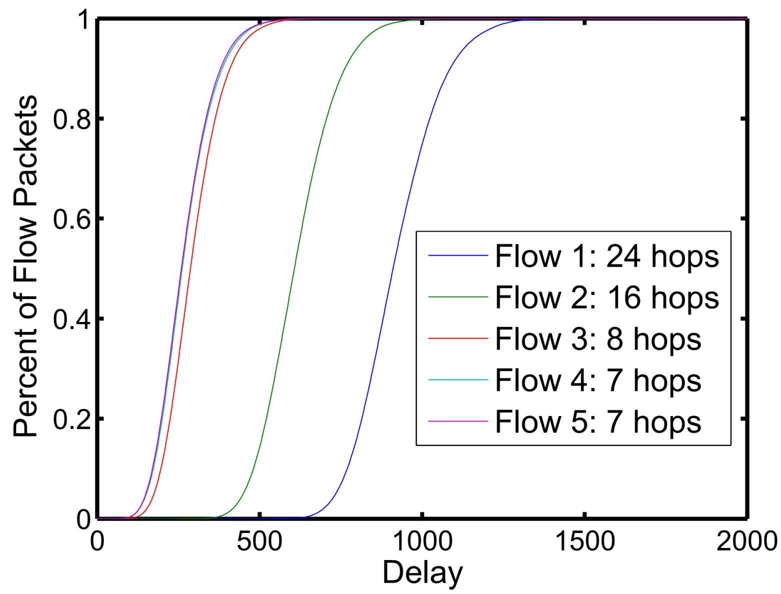


Figure 6.6: CDF of individual flow packet delay.

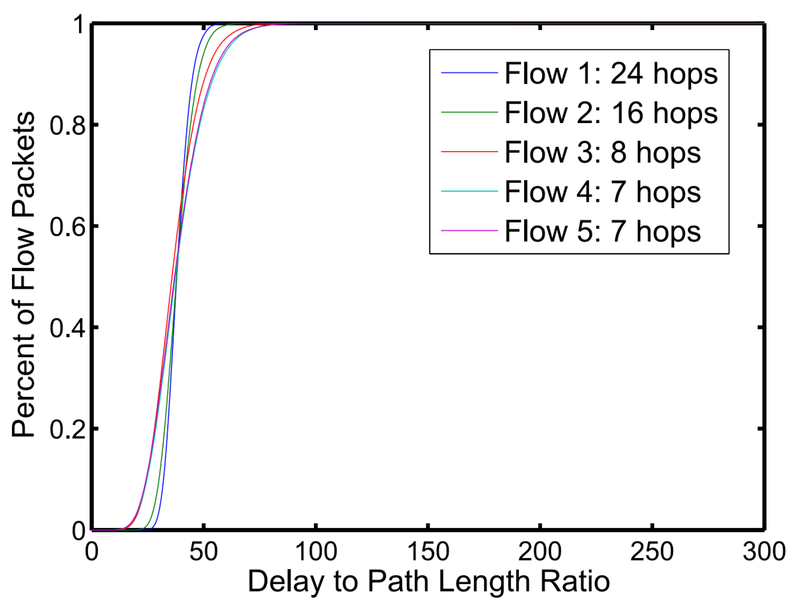


Figure 6.7: CDF of individual flow packet delay to path length ratio.

6.9.4 Average Queue Size

With the same setting as that of the previous set of experiments in Section 6.9.3, we observe the average backlogs for the following three representative types of flows: high-rate, medium-rate, and low-rate flows. The high-rate flow is close to saturating the stability condition in Inequality 6.2, while the medium-rate flow injects packets with half of the high rate, and the low-rate flow has an even smaller rate. Each of the flows has a path of at least 25 hops. Figure 6.8 shows the average queue sizes as a function of hop number. The average backlog of each flow stays much flatter than a quadratic curve in the number of hops; the higher the flow rate, the higher the backlog. We note that it is hard in general to prove a tight bound on average queue sizes and thereafter the average delays using Lyapunov drift techniques, which appears more efficient in the use of proving stability. For example, one can only get a delay bound exponential in path length by applying Lyapunov-drift-based analysis. Devising more efficient techniques stands as a major challenge to better understand scheduling in multi-hop wireless networks. It is necessary to develop problem-specific new analysis techniques towards this end, for example, in this chapter, the queueing reduction technique to derive the quadratic delay bounds for the random-access scheduling scheme.

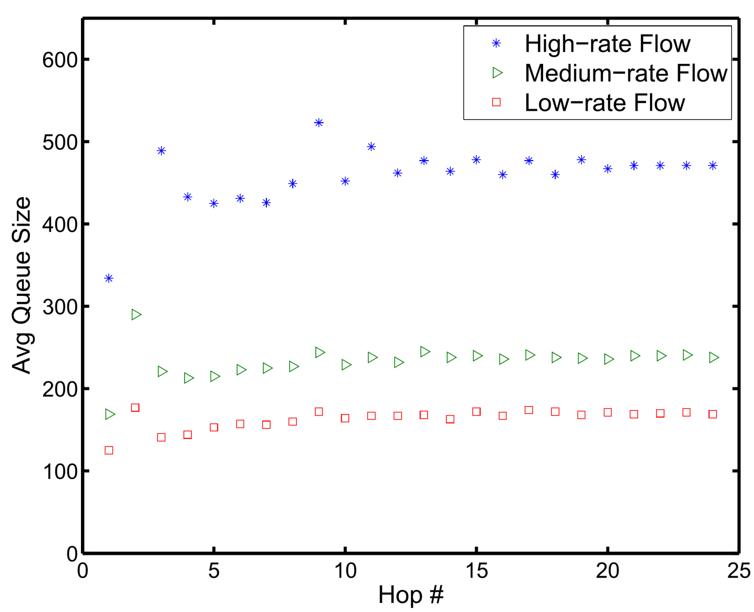


Figure 6.8: Average queue size with no transmission threshold.

6.10 Chapter Summary and Discussion

Characterizing delay-throughput tradeoffs and bounds is a fundamental problem in wireless networks, with numerous applications. In this chapter, we develop a theoretical framework to rigorously bound this tradeoff and provably approximate the maximum throughput with given per-session delay requirements. The performance guarantees we prove are worst case bounds, and are likely to be better for specific instances. Extending these techniques to bound the average per-session delay with additional fairness constraints is a very challenging open problem. Bounding per-session end-to-end delay in general is an open and difficult problem. New progress on this will likely enable our DCTM framework to work with a larger class of scheduling schemes, for example the maximal scheduling [106] and random-access maximal scheduling [49]. The simple random access scheduling scheme suffers from high delays for low rate vectors, and cannot give good bounds if heterogeneous throughput rates are required. Combining both the random access and EDF-type scheduling schemes might be one way to address this problem.

Chapter 7

Throughput Maximization for Secondary-users with Delay Guarantees in Cognitive Networks under the Physical Interference Model

7.1 Preliminaries and Definitions

The network model, traffic model, queueing model and related definitions are described in Chapter 4.

7.1.1 Revisiting Rate-based Random-access Scheduling RB-Sched under the Graph-based Interference Model

RB-SCHED involves the following process: at each time slot t, each link l stochastically makes channel access attempt with a specific probability $p_l(t)$ (known as *channel access probability*) when $Q_l(t) > 0$; if link l decides to transmit, it will choose a flow f associated with the link with probability $p(l, f)$, defined below. If no collision happens, it will result in successful data transmission, with an ACK packet sent backward at the end of the slot; otherwise, the packet will stay in the queue for the next transmission service. Note that the collision accounts for transmission of both directions on links, due to the definition of interference and the interference sets. In order to simplify our presentation, we ignore the complexity of handling ACKs, which can be done in the same manner as the flows for the sessions, *i.e.*, by reserving a constant fraction of the rate on every link— this only alters our approximation bounds by a constant factor. We focus on *synchronous* random-access scheduling, where all slots are of the same length. A random-access scheduling scheme (referred to as RB-Sched) has been used in [112] with the channel access probability for link l at time t as

$$p_l(t) = 1 - \exp\left(e \sum_{f:\, l \in L(f)} \lambda(f)/(1-\epsilon)\right), \tag{7.1}$$

where $\epsilon \in (0,1)$ denotes a *rate slackness parameter* which can be set before system initiation as a constant. For each transmission on l, $Q_{l,f}$'s packets get serviced with probability

$$p(l,f) = \frac{\lambda(f)}{\sum_{f':\, l \in L(f')} \lambda(f')}.$$

In [112] it is proved that RB-Sched is stable if

$$\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_{f:\, l' \in L(f)} \lambda(f) \leq \frac{1-\epsilon}{e}, \ \forall l \in L. \tag{7.2}$$

Note that RB-Sched is based only on the local rate information for each link. Inequality (7.2) and Theorem 4.1 defines a throughput region within a factor of $1/(e\mathcal{K})$ of Λ^{OPT}; note that $\mathcal{K}$ is usually a constant in most graph-based interference models.

7.2 Delay Metric and Problem Definition

The delay metric on focus is average delay, which can be deduced by Little's law in a stable system. We study the average end-to-end delay: (1) for each flow f, as the average time for packets to reach the destination following the flow path; (2) for each session c, as the average time for packets to reach the destination $t(c)$; and (3) for the entire network, as the average over all sessions.

We formally describe the problem we study in this chapter. Given a multi-hop wireless network represented by a graph $G = (\mathcal{V}, L)$, we have a set of primary users and a set of secondary users. There are a set $\mathcal{C}^p$ of on-going primary sessions with given routes and mean traffic rates for the primary users, and a set $\mathcal{C}^s$ of secondary session requests for the secondary users with a target delay $\Delta(c)$ (in number of time slots) for each session $c \in \mathcal{C}^s$. The main goal is to find a stable rate vector $\lambda()$ and routes, and a scheduling and power control scheme for all the secondary sessions to maximize the total achievable rate $\sum_c \lambda(c)$. At the same time, we ensure that for each session $c \in \mathcal{C}^s$, the per-packet delay is at most $\Delta(c)$; most importantly, we ensure that the throughput rates for all the existing primary sessions are not affected. Let OPT denote the maximum total throughput $\sum_c \lambda(c)$ for $\lambda() \in \Lambda^{OPT}$ with the freedom of arbitrary control across traffic rates, routing, scheduling and power control. Let $OPT\,(\Delta())$ denote the maximal total rate $\sum_c \lambda(c)$ that is feasible under these (delay) constraints.

As discussed in [112], the problem without the presence of primary users under a graph-based interference model is already computationally hard in general, and our focus is on approximation algorithms. In particular, we develop polynomial-time bi-criteria approximation algorithms; we say an algorithm that computes a rate vector $\lambda()$ gives a (β_1, β_2)-approximation if the total throughput rate guaranteed is at least $\beta_1 OPT\,(\Delta())$, while the delays are at most $\beta_2 \Delta(c)$ for each session. Note that these are worst case approximation guarantees, which hold for every problem instance.

7.3 Rate-based Random-access Scheduling RB-Sched-SINR under the Physical Interference Model

RB-SCHED-SINR is a $g(L)$-augmented version of RB-SCHED over link set L for scheduling in SINR setting, with the modifications listed below this paragraph. A slot under RB-SCHED extends to a super-slot under RB-SCHED-SINR, where RB-SCHED is performed over links in L_i at the ith slot of a super-slot. We do not distinguish between primary or secondary traffic. Below lists our modifications based on $g(L)$-augmented RB-SCHED:

(1) **Channel-access Probability.** We reset the probability of accessing the channel for any link l to

$$p_l(t) = 1 - e^{-eg(L)\lambda(l)/(1-\epsilon)},$$

where $\epsilon > 0$ is a constant that can be arbitrarily small.

(2) **Power Assignment.** The only constraint on power assignment is that, the power values

for transmission on any two links in the same set L_i are at most a constant factor away from each other. This incorporates uniform and linear power assignment schemes as special cases. For simplicity, we assume links in the same link set L_i use the same transmission power P_i. Removing this assumption may affect the results by no more than a constant factor.

7.3.1 Sufficient Condition for Stability under RA-Sched-SINR

Theorem 7.1 (Sufficient condition for stability)**.** RB-SCHED-SINR *guarantees queue-stability of the system under arrival processes that satisfy*

$$\forall l \in L, \sum_{l' \in \hat{\mathcal{I}}(l)} \lambda(l') \leq \frac{1-\epsilon}{e \cdot \Gamma \cdot g(L)}, \tag{7.3}$$

where $2 < \Gamma < 3$ *is a constant whose exact value will be determined later.*

Lemma 7.2. *Let* T *be an arbitrary super-slot with* $g(L)$ *slots. Under* RB-SCHED-SINR*, the network on* L *is stable if for any link* $l \in L_i$*,*

$$\mathbb{E}\{\sum_{t \in T} \mu_l(t)\} > g(L)\lambda(l), \forall T. \tag{7.4}$$

In the rest of the section, we assume Inequality (7.3) holds, and prove Lemma 7.2. Intuitively, Lemma 7.6 approximately suggests that the mean service rates surpass the mean arrival rates at the sources of flows. However, as stated in Section 6.5, to analyze the stability of the system is difficult for a multi-hop setting because (i) the exogenous arrival processes are general and makes the departure process hard to be analyzed and (ii) the endogenous arrival processes are more complicated due to the inter-dependency among interfering links. We first analyze the average service rate on each link of RB-SCHED-SINR under the physical interference model in Section 7.3.2, and then in Section 7.3.3 we derive the finite delay upper-bounds independent of network size n using the queueing system reduction technique developed in Section 6.5 of Chapter 6, providing a strong evident for the stability of a multi-hop network under RB-SCHED-SINR with the traffic subject to Inequality (7.3).

7.3.2 Service Rate

Let $h_l(t)$ denote if link l transmits at time t.

Proposition 7.3. *The expected number of active (or transmitting) links in $\hat{\mathcal{I}}(l)$ of any $l \in L$ at any time t is $\leq 1/\Gamma$.*

Proof. Let $N_l(t)$ denote the number of transmitting links in $\hat{\mathcal{I}}(l)$ at t; recall that $h_l(t) \in \{0,1\}$ indicates if l chooses to transmit at time t.

$$\begin{aligned}
\mathbb{E}\{N_l(t)\} &= \sum_{l' \in \hat{\mathcal{I}}(l)} \mathbb{E}\{h_{l'}(i)\} = \sum_{l' \in \hat{\mathcal{I}}(l)} p(l') \\
&= \sum_{l' \in \hat{\mathcal{I}}(l)} (1 - e^{-eg(L)\lambda(l')/(1-\epsilon)}) \\
&\leq \sum_{l' \in \hat{\mathcal{I}}(l)} eg(L)\lambda(l')/(1-\epsilon) \qquad (7.5) \\
&= eg(L)/(1-\epsilon) \sum_{l' \in \hat{\mathcal{I}}(l)} \lambda(l') \\
&\leq 1/\Gamma. \qquad (7.6)
\end{aligned}$$

Inequality (7.5) holds because

$$e^{-x} \geq 1 - x \Leftrightarrow x \geq 1 - e^{-x}, \forall x \geq 0.$$

□

Proposition 7.4. *We say two links k and k' are "independent" if they are not in each other's interference set. The distance between either end of k and either end of k' is at least $(\gamma - 2)d_i$.*

Lemma 7.5. *Let t be a time slot when links from L_i are scheduled. The probability for any $l \in L_i$ to make a successful transmission when there is no interference from $\mathcal{I}(l)$ is*

$$Prob\Big(u_l(t) = 1 \Big| \sum_{l' \in \mathcal{I}(l)} h_{l'}(t) = 0\Big) \geq p_l(t)(1 - 1/\Gamma).$$

Proof. We first partition the plane into concentric rings via a similar technique to that in [25, 103], and then by solving a set cover problem we develop an upper-bound on the minimum number of greater interference sets ($\hat{\mathcal{I}}$) that covers all the links that touches a ring, so that with reference to Proposition 7.3, we can bound the expected total interference from all the rings outside of $\mathcal{I}(l)$.

We partition the plane into rings all centered at $r(l)$, each of width $(\gamma - 2)d_i$. Let $R(j)$ denote the jth ring, which contains every transmitter node u of a link in L_i, that satisfy $j(\gamma-2)d_i \leq d(u, r(l)) < (j+1)(\gamma-2)d_i$, where $j = 1, 2, \ldots$. When $j = 0$, $R(0)$ corresponds to a disk of radius $(\gamma - 2)d_i$, within which the nodes all belong to links in $\hat{\mathcal{I}}(l)$ due to Proposition 7.4. We say that link $k \in R(j)$, if either $x(k)$ or $r(k)$ falls in $R(j)$. Let the total number of active (or transmitting) nodes in $R(j)$ at t be $N_{R(j)}(t)$. Since each link attempts to access the channel and transmit independently, and each link can have at most one active node (either sending data from transmitter node or sending ACK from receiver node), $N_{R(j)}(t)$ may fluctuate during t; yet, we have that during t, $\mathbb{E}\{N_{R(j)}(t)\} \leq \sum_{k\in R(j)} p_k(t)$.

Let $L^{cov}(j)$ be a minimal set of links in $R(j)$, such that

(1) $\bigcup_{k\in L^{cov}(j)} \hat{\mathcal{I}}(k)$ covers all the links in $R(j)$, and

(2) all the links in L^{cov} do not interfere with each other.

By Proposition 7.3,

$$\mathbb{E}\{N_{R(j)}(t)\} \leq \sum_{k\in L^{cov}(j)} \sum_{k'\in\hat{\mathcal{I}}(k)} p_{k'}(t) \leq |L^{cov}(j)|/\Gamma. \tag{7.7}$$

$L^{cov}(j)$ is a set of pairwise independent links in $R(j)$. Now we pick a node set $N^{cov}(j)$ that covers all links in $L^{cov}(j)$, such that each link has one and only one node in $N^{cov}(j)$, and all the nodes in $N^{cov}(j)$ fall in $R(j)$. By Proposition 7.4, disks of radius $(\gamma - 2)d_i/2$ centered at each of the nodes in $N^{cov}(j)$ do not overlap. All these disks are contained in an extended ring $R'(j)$ of $R(j)$, with extra width of $(\gamma - 2)d_i/2$ at each side of $R(j)$.

Therefore, $|L^{cov}(j)| = |N^{cov}(j)| \leq 2^3(2j + 1)$ due to packing property. That yields from Inequality (7.7) that

$$\mathbb{E}\{N_{R(j)}(t)\} \leq 2^3(2j + 1)/\Gamma.$$

Since on some links it may be the transmitter nodes that are transmitting data and on others it may be the receiver node that are transmitting ACKs, the interference observed can vary in a time slot. Let $I_j^x(t)$ and $I_j^r(t)$ denote the maximum total interference from the jth ring to $x(l)$ and $r(l)$ respectively at t. By noticing that $x(l)$ is on the circle centered at $r(l)$ with radius $d(l)$, we have

$$\begin{cases} I_j^x(t) \leq N_{R(j)}(t) \cdot \frac{P_i}{(j(\gamma-2)d_i - d(l))^\alpha}; \text{ and} \\ I_j^r(t) \leq N_{R(j)}(t) \cdot \frac{P_i}{(j(\gamma-2)d_i)^\alpha}. \end{cases}$$

Then, define $I_j(t) \triangleq N_{R(j)}(t) \cdot \frac{P_i}{(j(\gamma-3)d_i)^\alpha}$. Therefore, $I_j^x(t) \leq I_j(t)$ and $I_j^r(t) \leq I_j(t)$ at the same time.

Let $I^x(t) \triangleq \sum_{j=1}^{\infty} I_j^x(t)$ and $I^r(t) \triangleq \sum_{j=1}^{\infty} I_j^r(t)$ denote the maximum total interference from outside of $\mathcal{I}(l)$ respectively. Let $I(t) = \sum_{j=1}^{\infty} I_j(t)$; then, $I(t) \geq \max\{I^x(t), I^r(t)\}$.

$$\begin{aligned} &\mathbb{E}\{I(t)\} \leq \mathbb{E}\{\sum_{j=1}^{\infty} I_j(t)\} = \sum_{j=1}^{\infty} \mathbb{E}\{I_j(t)\} \\ =&\sum_{j=1}^{\infty} \frac{2^3}{\Gamma} P_i(\gamma-3)^{-\alpha} d_i^{-\alpha} \frac{2j+1}{j^\alpha} \\ \leq&\frac{3 \cdot 2^3}{\Gamma} P_i(\gamma-3)^{-\alpha} d_i^{-\alpha} \frac{\alpha-1}{\alpha-2}. \end{aligned}$$

Recall that $s_l(t) \in \{0,1\}$ indicates whether link l chooses to access the channel at time t. Let J_1 denote the event that "$s_l(t) = 1$," J_2 the event that "$\sum_{l' \in \mathcal{I}(l)} s_{l'}(t) = 0$," and J_3 the event that "$I(t) < \Gamma \cdot \mathbb{E}\{I(t)\}$." According to Markov's Inequality:

$$Prob\,(J_3) = 1 - Prob\big(I(t) \geq \Gamma \cdot \mathbb{E}\{I(t)\}\big) \geq 1 - 1/\Gamma.$$

With J_1, J_2, and J_3 being true at the same time, the SINR values at $x(l)$ and $r(l)$ are at least

$$\frac{P_i d^{-\alpha}(l)}{\Gamma \mathbb{E}\{I(t)\} + N} \geq \frac{P_i d_i^{-\alpha}}{3 \cdot 2^3 \sigma^\alpha P_i (\gamma-3)^{-\alpha} d_i^{-\alpha} \frac{\alpha-1}{\alpha-2} + N} \geq \beta.$$

That means both data transmission from $x(l)$ and ACK transmission from $r(l)$ will be successful, *i.e.*, $u_l(t) = 1$.

Therefore, $Prob\,\big(u_l(t) = 1 \big| J_2\big) \geq Prob\,\big(J_1 \cap J_3 \big| J_2\big)$ leading to the statement. □

Lemma 7.6. *Let t be a time slot when only links from L_i are scheduled for any $i = 1, 2, \ldots, g(L)$. The expected service rate for any $l \in L_i$ is*

$$\mathbb{E}\{\mu_l(t)\} \geq g(L)\lambda(l)/(1-\epsilon). \tag{7.8}$$

Proof. Let J_4 denote the event that "$\sum_{l' \in \mathcal{I}(l)} h_{l'}(t) = 0$," *i.e.*, all links in $\mathcal{I}(l)$ are silent. By applying Lemma 7.5, any link $k \in \mathcal{I}(l)$ makes a successful transmission with probability

$$
\begin{aligned}
&Prob\{u_k(t) = 1\} \geq Prob\,(u_k(t) = 1 \cap J_4) \\
\geq &Prob\,\big(u_k(t) = 1 \,\big|\, J_4\big) \cdot Prob(J_4) \\
\geq &p_l(t)(1 - 1/\Gamma) \prod_{l' \in \mathcal{I}(l)} (1 - p_{l'}(t)) \\
= &(1 - 1/\Gamma)(1 - e^{-eg(L)\lambda(l)/(1-\epsilon)}) \prod_{l' \in \mathcal{I}(l)} e^{-eg(L)\lambda(l')/(1-\epsilon)} \\
= &(1 - 1/\Gamma)(1 - e^{-eg(L)\lambda(l)/(1-\epsilon)}) e^{-\sum_{l' \in \mathcal{I}(l)} eg(L)\lambda(l')/(1-\epsilon)} \\
= &(1 - 1/\Gamma)(1 - e^{-eg(L)\lambda(l)/(1-\epsilon)}) e^{eg(L)\lambda(l)/(1-\epsilon)-1} \\
= &(1 - 1/\Gamma)(e^{eg(L)\lambda(l)/(1-\epsilon)} - 1) e^{-1/\Gamma} \\
\geq &(1 - 1/\Gamma) e^{1-1/\Gamma} g(L)\lambda(l)/(1-\epsilon).
\end{aligned}
$$

We notice that $(1 - 1/\Gamma)e^{1-1/\Gamma}$ is a monotonically increasing function of Γ. Hence, in order for Inequality (7.8) to hold, we only need to find a value for Γ such that $(1-1/\Gamma)e^{1-1/\Gamma} \geq 1$. This reminds us of the Lambert W Function, which is the inverse of function $f(x) = xe^x$. Although larger Γ increases the probability of success for transmissions, it at the same time shrinks the throughput region as in Inequality (7.3). Therefore, we are interested in finding a minimal Γ such that $(1 - 1/\Gamma)e^{1-1/\Gamma} \geq 1$. By solving for $f(x) = 1$, we obtain $2.31 < \Gamma \approx 2.3105 < 2.32$. Taking $\Gamma = 2.32$ yields $Prob\{u_k(t) = 1\} \geq g(L)\lambda(l)/(1 - \epsilon)$, leading to the statement. □

7.3.3 Delay Upper-bounds for RA-Sched-SINR

We derive the delay upper-bounds for flows with a given feasible average rate vector $\lambda()$ that satisfies the constraints in Inequality (7.3).

Theorem 7.7. *For a rate vector $\lambda()$ that satisfies Inequality (7.2),* RB-Sched-SINR *ensures that (i) the system is stable, implying a throughput region of $O(\frac{1}{g(L)})\Lambda^{OPT}$; (ii) the average delay for each flow f is $O\left(|L(f)|^2/(\lambda(f))^2\right)$; and (iii) the average network delay is $O\left(\sum_{f \in \mathcal{F}} |L(f)|^2 \Big/ \left(\sum_{f \in \mathcal{F}} \lambda(f) \min_{f \in \mathcal{F}}\{\lambda(f)\}\right)\right)$.*

Under RB-Sched-SINR, each super-slot consists of $g(L)$ slots and in the jth slot, only links from L_j are scheduled. For a link $l \in L_j$, we say l *participates* at time t, if t is the jth slot in a super-slot; otherwise, l does not participate.

Now we derive a lower bound on the expected service rate $\mu_{l,f}(T)$ for the link-flow pair (l, f). For notational simplicity, we use $x(l, f) = g(L)\lambda(f)/(1 - \epsilon)$, and $x(l) = \sum_{f \in \mathcal{F}(l)} x(l, f)$.

Lemma 7.8. *Let t be a time slot when a link l participates, and let t' be a time slot when l does not participate. The expected service rates under* RB-SCHED-SINR *for any $l \in L_i$ at t and t' are*

$$\mathbb{E}\left\{\mu_{l,f}(t)\right\} \geq x(l, f), \tag{7.9}$$

and

$$\mu_{l,f}(t') = 0. \tag{7.10}$$

Proof. Lemma 7.6 implies the following lower bound on the expected service rate for a link-flow pair (l, f):

$$\mathbb{E}\left\{\mu_{l,f}(t)\right\} \geq p(l, f)\mathbb{E}\left\{\mu_l(t)\right\} \geq g(L)\lambda(f)/(1 - \epsilon) = x(l, f).$$

$\mu_{l,f}(t') = 0$ because $l \in L_i$ remains silent at time t'. □

The idea for the proof of the above theorem is that due to the properties of random-access scheduling, each flow can be viewed in "isolation" as a tandem system, with lower bounds on the expected service rate of $\mu(l, f)$ that only depend on $x(l, f)$ for each logical queue $Q_{l,f}$, as shown in Equation (7.9). This technique is developed in Chapter 6, and is independent of interference model; here, we provide all the necessary steps following those in Section 6.5 and reuse some of the lemmas from Section 6.5, while we accommodate the additional factor $g(L)$ in the analysis because of the scheduling mechanism that activate the $g(L)$ link classes separately in time.

Let the triple $(Q(), a(), \mu())$ denote a queueing system. From now on till the end of this section, we use R to denote the queueing system under the original RB-SCHED-SINR scheduling scheme, with the queueing model and the exogenous arrival processes described in Section 7.1. We put R at superscript to denote the quantities of the corresponding system.

We now consider the queues for a specific flow f: $\{Q^R_{i_f,f}\}$, $i \in \{1, 2, \ldots, |L(f)|\}$, as a series of tandem queues, and derive delay bounds. Our proof involves two "reductions", which progressively lead to a simpler queueing system with Bernoulli arrival and service processes for the non-source queues, with delays no smaller than those of $\{Q^R_{i_f,f}\}$; additionally, the second queueing system we construct has an increasing sequence of service rates, allowing us to derive end-to-end delay bounds. We start with the following intuitive lemma, which can be proved by induction on t.

Lemma 7.9 (from Lemma 6.5)**.** *Let R' and R'' be two tandem queueing systems with the same initial states and the same arrival processes but $\mathbb{E}\left\{\mu^{R'}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\mu^{R''}_{i_f,f}(t)\right\}$, for each flow f and link i_f, at time t. Then, for each f,t*

$$\mathbb{E}\Big\{\sum_{i=1}^{|L(f)|} Q^{R''}_{i_f,f}(t)\Big\} \geq \mathbb{E}\Big\{\sum_{i=1}^{|L(f)|} Q^{R'}_{i_f,f}(t)\Big\}.$$

(1) **Reduction 1**. We reduce the basic queueing system R for flow f to another tandem system R_1, such that for all $i \in [1, |L(f)|]$, at a time slot t when link i_f participates, the service rate $\mu^{R_1}_{i_f,f}$ for the queue $Q^{R_1}(i_f, f)$ is subject to a Bernoulli distribution with

$$\mathbb{E}\left\{\mu^{R_1}_{i_f,f}(t)\right\} = g(L)(\lambda(f) + \frac{i\varepsilon(f)}{|L(f)|}) \leq x^R(i_f, f) \leq \mathbb{E}\left\{\mu^R_{i_f,f}(t)\right\},$$

where $\varepsilon(f) = \epsilon\lambda(f)/(1-\epsilon)$, and at a time slot t' when link i_f does not participate, $\mu^{R_1}_{i_f,f}(t') = 0$; the exogenous arrival rates remain the same.

Then, Lemma 7.9 implies that $\mathbb{E}\left\{\sum_{i=1}^{|L(f)|} Q^{R_1}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\sum_{i=1}^{|L(f)|} Q^R_{i_f,f}(t)\right\}$ for all f, t. Note that whether the reduced system is using wireless medium no longer matters.

(2) **Reduction 2**. We reduce the system R_1 to another queueing system R_2 in the same way as that in Section 6.5 of Chapter 6, so that the arrival process for each non-source queue is also Bernoulli and we have the following lemma.

Lemma 7.10 (from Lemma 6.6)**.** *At every time slot t,*

$$\mathbb{E}\left\{\sum_{i=1}^{|L(f)|} Q^{R_2}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\sum_{i=1}^{|L(f)|} Q^{R_1}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\sum_{i=1}^{|L(f)|} Q^{R}_{i_f,f}(t)\right\}.$$

(3) **Queueing analysis for R_2**. The queueing analysis for R_2 follows the same process as that in Section 6.5. Here we provide details of how we accommodate the factor $g(L)$

while calculating and bounding the service rates and arrival rates for R_2. The fact that the arrival and service processes of each $Q^{R_2}_{i_f,f}$ are subject to Bernoulli distribution, allows us to perform the isolated queueing analysis for each $Q^{R_2}_{i_f,f}$ in isolation. For any link i_f where $i = 1, 2, \ldots, |L(f)|$, at a time slot t when link i_f participates,

$$\mathbb{E}\left\{\mu^{R_2}_{i_f,f}(t)\right\} = \mathbb{E}\left\{\mu^{R1}_{i_f,f}(t)\right\} = g(L)\big(\lambda(f) + \frac{i\varepsilon(f)}{|L(f)|}\big),$$

and at a time slot t' when i_f does not participate, $\mu^{R_2}_{i_f,f}(t) = \mu^{R_1}_{i_f,f}(t) = 0$.

Next, we perform Lyapunov drift analysis to derive an upper-bound on the queue size of each $Q^{R_2}_{i_f,f}$. Refer to [43, 106] for the details of this approach. We define the Lyapunov function as

$$\mathbb{L}\left(Q^{R_2}_{i_f,f}(t)\right) \triangleq \left(Q^{R_2}_{i_f,f}(t)\right)^2.$$

The $g(L)$-step Lyapunov drift is then defined as:

$$\triangle^{(g(L))}_{Q}\left(Q^{R_2}_{i_f,f}(t)\right) \triangleq \mathbb{E}\left\{\mathbb{L}\left(Q^{R_2}_{i_f,f}(t+g(L))\right) - \mathbb{L}\left(Q^{R_2}_{i_f,f}(t)\right)\Big|Q^{R_2}_{i_f,f}(t)\right\}.$$

The $g(L)$-step queue-evolution can be expressed as

$$\begin{aligned}
&Q_{i_f,f}(t+g(L)) \\
=&Q_{i_f,f}(t) - \sum_{\tau=t}^{t+g(L)-1} d_{i_f,f}(\tau) + \sum_{\tau=t}^{t+g(L)-1} a_{i_f,f}(\tau) \\
\leq&\left[Q_{i_f,f}(t) - \sum_{\tau=t}^{t+g(L)-1} \mu_{i_f,f}(\tau)\right]^+ + \sum_{\tau=t}^{t+g(L)-1} a_{i_f,f}(\tau),
\end{aligned} \tag{7.11}$$

By referring to Inequality (7.11) and Lemma 4.3 of [43], we obtain

$$\begin{aligned}
&\triangle^{(g(L))}_{Q}\left(Q^{R_2}_{i_f,f}(t)\right) \\
\leq&\mathbb{E}\Bigg\{\Bigg(\sum_{\tau=t}^{t+g(L)-1} \mu^{R_2}_{i_f,f}(\tau)\Bigg)^2 + \Bigg(\sum_{\tau=t}^{t+g(L)-1} a^{R_2}_{i_f,f}(\tau)\Bigg)^2\Bigg|Q^{R_2}_{i_f,f}(t)\Bigg\} - \\
&2Q^{R_2}_{i_f,f}(t)\sum_{\tau=t}^{t+g(L)-1}\mathbb{E}\left\{\left(\mu^{R_2}_{i_f,f}(\tau) - a^{R_2}_{i_f,f}(\tau)\right)\Big|Q^{R_2}_{i_f,f}(t)\right\}.
\end{aligned} \tag{7.12}$$

Further, since $0 \leq \sum_{\tau=t}^{t+g(L)-1} \mu_{i_f,f}^{R_2}(\tau) \leq 1$, we have

$$\mathbb{E}\left\{\left(\sum_{\tau=t}^{t+g(L)-1} \mu_{i_f,f}^{R_2}(\tau)\right)^2\right\} \leq \mathbb{E}\left\{\sum_{\tau=t}^{t+g(L)-1} \mu_{i_f,f}^{R_2}(\tau)\right\} \leq g(L)/(e\Gamma),$$

and $\sum_{\tau=t}^{t+g(L)-1} \mathbb{E}\left\{\mu_{i_f,f}^{R_2}(\tau) - a_{i_f,f}^{R_2}(\tau)\right\} = g(L)\varepsilon(f)/|L(f)|$.

Additionally, for $i = 1$,

$$\mathbb{E}\left\{\left(\sum_{\tau=t}^{t+g(L)-1} a_{1_f,f}^{R_2}(\tau)\right)^2\right\} \leq g(L)A^{(2)} + (g(L)\lambda(f))^2 \leq g(L)A^{(2)} + 1/(e\Gamma);$$

when $i > 1$, since $\sum_{\tau=t}^{t+g(L)-1} a_{i_f,f}^{R_2}(\tau) = \sum_{\tau=t}^{t+g(L)-1} \mu_{(i-1)_f,f}^{R_2}(\tau) \leq 1$,

$$\mathbb{E}\left\{\left(\sum_{\tau=t}^{t+g(L)-1} a_{i_f,f}^{R_2}(\tau)\right)^2\right\} \leq \mathbb{E}\left\{\sum_{\tau=t}^{t+g(L)-1} a_{i_f,f}^{R_2}(\tau)\right\} \leq g(L)/(e\Gamma).$$

For $i = 1$, Inequality (7.12) can be then rewritten as

$$\triangle_Q^{(g(L))}\left(Q_{1_f,f}^{R_2}(t)\right) \leq g(L)\big(A^{(2)} + \frac{1}{e\Gamma} - \frac{2\varepsilon(f)}{|L(f)|}Q_{1_f,f}^{R_2}(t)\big).$$

For $i > 1$, Inequality (7.12) can be then rewritten as

$$\triangle_Q^{(g(L))}\left(Q_{i_f,f}^{R_2}(t)\right) \leq g(L)\big(\frac{2}{e\Gamma} - \frac{2\varepsilon(f)}{|L(f)|}Q_{i_f,f}^{R_2}(t)\big).$$

From Theorem 1 in [106] which deduces an inequality from the Lyapunov drift, the mean backlog $\overline{Q}_{i_f,f}^{R_2}$ is

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{Q_{i_f,f}^{R_2}(\tau)\right\} \leq \frac{1+e\Gamma A^{(2)}}{2e\Gamma\varepsilon(f)}|L(f)| \text{ for } i = 1,$$

and

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{Q_{i_f,f}^{R_2}(\tau)\right\}\le\frac{1}{e\Gamma\varepsilon(f)}|L(f)|\text{ for }i>1.$$

Therefore, the sum of mean backlogs for flow f is

$$\sum_{i=1}^{|L(f)|}\overline{Q}_{i_f,f}^{R_2}(t)\le\frac{2|L(f)|-1+e\Gamma A^{(2)}}{2e\Gamma\varepsilon(f)}|L(f)|\le\frac{\max\left\{2,e\Gamma A^{(2)}\right\}}{2e\Gamma\varepsilon(f)}|L(f)|^2.$$

Corollary 7.11. *The systems R, R_1 and R_2 are all stable.*

(4) **Average end-to-end delay bound**. By Little's Law, the average delay for flow f's packets is

$$\overline{D}^R(f)=\sum_{i=1}^{|L(f)|}\overline{Q}_{i_f,f}^{R}\Big/\lambda(f)\le\sum_{i=1}^{|L(f)|}\overline{Q}_{i_f,f}^{R_2}\Big/\lambda(f)\le\frac{\max\left\{2,e\Gamma A^{(2)}\right\}}{2e\Gamma\varepsilon(f)}\frac{|L(f)|^2}{\lambda(f)}.$$

The average network delay is

$$\overline{D}^R=\sum_{f\in\mathcal{F}}\sum_{i=1}^{|L(f)|}\overline{Q}_{i_f,f}^{R}\Big/\sum_{f\in\mathcal{F}}\lambda(f)\le\frac{\max\left\{2,e\Gamma A^{(2)}\right\}}{2e\Gamma\min_{f\in\mathcal{F}}\{\varepsilon(f)\}}\frac{\sum_{f\in\mathcal{F}}|L(f)|^2}{\sum_{f\in\mathcal{F}}\lambda(f)}.$$

Theorem 7.7 follows by substituting $\epsilon\lambda(f)/(1-\epsilon)$ for $\varepsilon(f)$.

7.3.4 Efficiency Ratio of RB-Sched-SINR

For RA-Sched-SINR, Theorem 7.1 and Theorem 4.1 together characterize a throughput region of nearly $\frac{1}{\mathcal{K}\cdot e\cdot\Gamma\cdot g(L)}$ of the optimum Λ^{OPT}. $\mathcal{K}$ in the SINR setting is the cardinality of a maximum independent set under optimum power assignment in $\mathcal{I}(l)$ of any link l. Due to Lemma 7.12, $\mathcal{K}$ is at most a constant. We conclude that the throughput region of RA-Sched-SINR is $\Omega(1/g(L))\Lambda^{OPT}$.

Lemma 7.12 (from Lemma 5.12)**.** *Under any scheduling and power assignment scheme, $\mathcal{K}$ is at most a constant in our SINR setting.*

7.4 Secondary-user Delay-constrained Throughput Maximization under the Physical Interference Model

The secondary-user delay-constrained throughput maximization (SU-DCTM) problem is to explore the opportunistic use of wireless channels for a set of secondary users under the presence of a set of primary users. With the existence of a set $\mathcal{C}^p$ of on-going primary sessions, The secondary users have a set $\mathcal{C}^s$ of secondary session requests and the goal of SU-DCTM has three folds:

(1) finding a stable rate vector $\lambda()$ for the secondary sessions to maximize the total achievable rate $\sum_{c\in\mathcal{C}^s}\lambda(c)$;

(2) ensuring that the per-packet delay for each secondary session $c\in\mathcal{C}^s$ is as close to $\Delta(c)$ as possible;

(3) the throughput rates for all the existing primary sessions are not affected.

In order to achieve the goal, we create a cross-layer optimization framework accounting for routing, scheduling, rate control and power control.

From Theorem 7.7, it follows that the average end-to-end delay bound for flow f is proportional to $|L(f)|^2$, and inversely proportional to $\lambda^2(f)$; the average network end-to-end delay bound is in proportion to $\sum_{f\in\mathcal{F}}|L(f)|^2$, and inversely in proportion to $\sum_{f\in\mathcal{F}}\lambda(f)$ and $\min_{f\in\mathcal{F}}\lambda(f)$. Therefore, in order to find a feasible rate vector $\lambda()$ that minimizes the delay guarantees, we need to construct flows with "high" rate (*i.e.*, to keep $\lambda(f)$ high) and "short" paths (*i.e.*, to make $|L(f)|$ low). In this section, we use terms "path" and "flow" interchangeably.

Given the delay constraint $\Delta(c)$ for each secondary session request $c\in\mathcal{C}^s$ (as defined in Section 7.1), we present a multi-commodity flow framework for choosing a rate vector $\lambda()$ and constructing a set of flows $\mathcal{F}''(c)$ for each session c, with the following properties (recall the definition of $OPT(\Delta())$ in Section 7.1).

Theorem 7.13. *(i) The rate vector $\lambda()$ resulting from our multi-commodity flow framework ensures that: $\sum_c\lambda(c)=\Omega\left(\frac{\log\log\Delta_m}{g(L)\log\Delta_m}\right)OPT\left(\Delta()\right)$. (ii) For each flow f in the set of flows $\mathcal{F}''(c)$ constructed for each session c if $\lambda(c)>0$, the rate $\lambda(f)=\Omega\left(\frac{\log\log\Delta_m}{g(L)\log\Delta_m}\right)$, and the path has a length of at most $2\Delta(c)$.*

Putting everything together: the total rate is "close" to $OPT\,(\Delta())$, and we have the following delay bounds.

Corollary 7.14. *Using the rate vector $\lambda()$ along with the random-access scheduling scheme described in Section 7.3.3, we ensure that, for each session c and each flow $f \in \mathcal{F}''(c)$, the average delay is $O\left(\left(\frac{g(L)\log\Delta_m}{\log\log\Delta_m}\Delta(c)\right)^2\right)$.*

Our algorithm involves construction of multi-commodity flows with constraints on the paths used; broadly, these constraints bound the sum of the "costs" of the links on the paths, which will be explained later with the LP formulation. Our approach employs an approximation algorithm, which selectively drops some cost-unfavorable sessions and maximizes the rates of the rest sessions. This has been first developed in Section 6.6 of Chapter 6; here, we provide all the necessary steps to consider the existence of the primary users, to ensure the SINR constraints are always satisfied and to accommodate the additional factor $g(L)$ in the analysis because of the scheduling mechanism that activate the $g(L)$ link classes separately in time. The concept of interference set we created in the context of SINR is key to the design and analysis of our optimization framework. Our algorithm is comprised of the following 4 steps: (1) calculation of existing channel occupation, (2) linear programming formulation, (3) path filtering, and (4) randomized rounding.

(Step 0) **Calculation of existing channel occupation**. For each link l, we define the channel occupation $E(l)$ as the total ongoing traffic rate of all sessions through any of the links in $\mathcal{I}(l) \cup \{l\}$. Then $E^p(l)$ denotes the existing channel occupation of primary sessions within l's proximity:

$$E^p(l) = \sum_{l' \in (\mathcal{I}(l) \cup \{l\})} \sum_{c \in \mathcal{C}^p} \sum_{f \in \mathcal{F}(c)} \lambda(l', f).$$

$E^s(l)$ denotes channel occupation of secondary sessions within l's proximity:

$$E^s(l) = \sum_{l' \in (\mathcal{I}(l) \cup \{l\})} \sum_{c \in \mathcal{C}^s} \sum_{f \in \mathcal{F}(c)} \lambda(l', f).$$

$E(l) = E^p(l) + E^s(l)$. The sufficient stability condition in Inequality (7.3) for RA-Sched-SINR can be rewritten as $E(l) \le \frac{1-\epsilon}{e\Gamma g(L)}$. We assume that there exists a constant $\epsilon' \in (\epsilon, 1)$, such that $E^p(l) \le \frac{1-\epsilon'}{e\Gamma g(L)}$. That means that $E^p(l)$ uses up at most a constant fraction of $\frac{1}{e\Gamma g(L)}$, so that $E^s(l)$ can be as large as $\frac{\epsilon'-\epsilon}{e\Gamma g(L)} = \Omega(\frac{1}{g(L)})$.

(Step 1) **Linear programming formulation**.

LP:

$$\max \quad \sum_{c\in\mathcal{C}^s} y(c)$$

$$\text{s.t.} \quad \forall c \in \mathcal{C}^s,\ y(c) = \sum_{f\in\mathcal{F}(c)} y(f) \tag{7.13a}$$

$$\forall c \in \mathcal{C}^s,\ \sum_{f\in\mathcal{F}(c)} y(f)cost(f) \le \Delta(c)y(c) \tag{7.13b}$$

$$\forall l, c \in \mathcal{C}^s,\ y(l,c) = \sum_{f\in\mathcal{F}(c):\ l\in L(f)} y(f) \tag{7.13c}$$

$$\forall l,\ \sum_{l'\in(\mathcal{I}(l)\cup\{l\})} \sum_{c\in\mathcal{C}^s} y(l',c) \le \frac{\epsilon'-\epsilon}{e\Gamma g(L)} \tag{7.13d}$$

$$\forall c \in \mathcal{C}^s, f\in\mathcal{F}(c),\ y(f) \ge 0 \tag{7.13e}$$

In the above formulation: (1) Constraints LP-(7.13a) and LP-(7.13c) represent path-based flow-conservation constraints. (2) LP-(7.13b) constrains the total path cost, which we use as a lower-bound on the average delay along a flow-path; in our case the cost function is chosen to be path length since end-to-end delay is lower bounded by the number of hops. (3) Congestion constraints in LP-(7.13d) ensure the stability under a random-access scheme for the cognitive network with the coexistence of primary and secondary users.

We write the optimal objective value of (LP) as $OPT^{LP}(\Delta())$. Since LP-(7.13b) serves as relaxed delay constraints and LP-(7.13d) may potentially compromise the optimal value of total flow by at most a factor of $g(L)$ due to Lemma 7.12, $OPT^{LP}(\Delta()) = \Omega\big(OPT(\Delta())/g(L)\big)$.

The above program may have exponentially many constraints because it is formulated using all the flow paths in $\mathcal{F}(c)$, which may include all viable paths in graph G. We reformulate the above LP into a polynomial sized one as follows:

(1) replacing Constraints LP-(7.13a) and LP-(7.13c) with for all $c \in \mathcal{C}^s$, $\sum_{l\in L_{out}(s(c))} y(l,c) = y(c)$ and $\sum_{l\in L_{in}(t(c))} y(l,c) = y(c)$, and flow-conservation constraints at all other nodes;

(2) replacing LP-(7.13b) with for all $c \in \mathcal{C}^s$, $\sum_{l\in L} y(l,c)cost(l) \le \Delta(c)y(c)$; and

(3) replacing LP-(7.13e) with for all $c \in \mathcal{C}^s, l \in L$, $y(l,c) \geq 0$.

Let $y^*()$ denote the optimum fractional solution to the above LP; $OPT^{LP}(\Delta()) = \sum_c y^*(c)$. Following standard techniques, *e.g.*, [6], this flow can be decomposed into path flows $y^*(f)$ in polynomial time, with a polynomial number of paths that have positive flow; let $\mathcal{F}^* = \{f : y^*(f) > 0\}$ be the set of flows with positive flow.

(Step 2) **Filtering**. We adopt the same path filtering method as that in Section 6.6 which is not affected by the $g(L)$ factor. For each connection $c \in \mathcal{C}^s$ we select paths (obtained from the solution to (LP)) with lengths of at most $2\Delta(c)$, and thus obtain a new flow vector $y'()$. As shown by Inequality (6.6), $\forall c,\ y'(c) \geq y^*(c)/2$. Let $\mathcal{F}' = \{f \in \mathcal{F} : y'(f) > 0\}$ be the set of flows with positive flow; for each $f \in \mathcal{F}'$, we have $cost(f) \leq 2\Delta(c)$.

(Step 3) **Randomized rounding**. In this step, we round the filtered solution to an integral solution to obtain Lemma 7.15.

Lemma 7.15. *After the randomized rounding step, we obtain a set $\mathcal{F}'' \subseteq \mathcal{F}^*$ of paths with positive rates, and a rate vector $\lambda()$, such that (1) for each $f \in \mathcal{F}''$, $\lambda(f) = \Omega\left(\log\log \Delta_m / \log \Delta_m\right)$; (2) $\sum_c \lambda(c)$ is $\Omega\left(\log\log \Delta_m / \log \Delta_m\right) OPT^{LP}(\Delta())$; and (3) the chosen paths incur "low" congestion; more precisely, for each link l, we have $\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_c \lambda(l',c) \leq \frac{\epsilon' - \epsilon}{e\Gamma g(L)}$, where $\lambda(l',c)$ is the rate on all paths $f \in \mathcal{F}''(c)$ such that $l' \in L(f)$.*

We first describe the sub-steps of the rounding stage and then discuss the proof of Lemma 7.15.

(1) **Pre-processing:** we partition paths into groups, formulate a minimax integer program (MIP) that minimizes maximum congestion and that chooses one path in each group with "large" flow rate, and formulate a relaxation of the minimax integer program with refined paths.
(2) **Randomized rounding:** we employ the techniques based on [131] to derive an approximate solution to the relaxed minimax integer program.
(3) **Post-processing:** we scale down the flow rates by a "reasonably small" factor, such that the congestion constraints LP-(7.13d) can be satisfied.

The above sub-steps follow the work flow of the rounding stage in Section 6.6. We minimize the impact of the SINR model and the $g(L)$-augmented scheduling, such that the main flow of analysis follows that in Section 6.6. The starting steps 3.1, 3.2 which bin-pack the paths and create an MIP formulation based the solution to the LP remain the same as those in Section 6.6 except for a simple change of a factor of $g(L)$. Steps 3.3, 3.4, 3.5 depends on the

SINR model and the concept of interference sets we created in the SINR setting, plus the factor of $g(L)$. We reuse the same rounding solution for Steps 3.6 as that in Section 6.6 and arrive at a vector $x'()$ in Lemma 7.16 which includes the factor $g(L)$. In Step 3.7, we scale the solution by a proper factor, such that the traffic vector is within the throughput region of RB-SCHED-SINR.

(Step 3.1) **Pre-processing: Bin-packing**. We partition $\mathcal{F}'$ into a sequence of k groups $\mathcal{F}'_1, \mathcal{F}'_2, \ldots, \mathcal{F}'_k$ of paths in the same way as that in Section 6.6, such that each group has a small total rate. Note that here $k = \Theta(g(L) \cdot OPT^{LP})$, since $\sum_{f \in \mathcal{F}'_{max}} y'(f) = \Theta(1/g(L))$ according to (LP).

(Step 3.2) **Pre-processing: MIP formulation**.

$$
\begin{aligned}
\textbf{MIP:} \quad \min \quad & w \\
\text{s.t.} \quad & \sum_{f \in \mathcal{F}'_i} z(f) = 1, \ \forall i = 1, \ldots, k && \text{(7.14a)} \\
& \sum_{l' \in \mathcal{I}(l) \cup \{l\}} \ \sum_{f \in \mathcal{F}' : l' \in L(f)} z(f) \leq w, \ \forall l && \text{(7.14b)} \\
& z(f) \in \{0, 1\}, \ \forall f \in \mathcal{F}' && \text{(7.14c)}
\end{aligned}
$$

(MIP) above formulates a minimax integer program that minimizes maximum congestion among all the interference sets. The formulation is the same as that in Section 6.6 except that the optimum objective value is impacted by a factor of $g(L)$. Constraints MIP-(7.14a) and MIP-(7.14c) let us choose one flow path from each set $\mathcal{F}'_i$, and assign flow rate of 1 to the paths chosen. Since $\sum_{f \in \mathcal{F}'_{max}} y'(f) = \Theta(1/g(L)$, the vector y', after suitable scaling of $O(g(L))$ is a feasible solution for the linear relaxation of (MIP); further, it will turn out that the objective value of an optimum fractional solution to (MIP) is $O(1)$. MIP-(7.14b) can be rewritten in an aggregate manner as $\mathbf{B}\vec{z} \leq \vec{w}$, where $\mathbf{B}$ is a $|L| \times |\mathcal{F}'|$ matrix. Intuitively, this integer program is hard to solve exactly, because matrix $\mathbf{B}$ can be dense and irregular, as a result of the facts that an interference set can be as large as L and that both of the set of flow paths in an optimal solution and the graph topology are non-controllable. In light of this, we perform the following path refinement (Step 3.3), constraint relaxation (Step 3.4) and MIP reformulation (Step 3.5) to approximately solve (MIP).

(Step 3.3) **Pre-processing: Path refinement**. Since the interference set of a link l in our SINR setting is defined based on a circular region centered at the sender of link l with a radius that is constant times $d(l)$, if there is a path $f \in \mathcal{F}'$ that uses more than a constant number, K_0, of links in $\mathcal{I}(l)$, we can always "short-cut" f into f' that uses at most K_0 such links, and does not violate any of the constraints of (LP).

(Step 3.4) **Pre-processing: Relaxation of congestion constraints**. The geometric property of the interference sets we define in the SINR setting helps reuse the logic and the basic work flow from Section 6.6 during the process of relaxation of congestion constraints in MIP-(7.14b). We have in Step 3.2 rewritten MIP-(7.14b) into $\mathbf{B}\vec{z} \leq \vec{w}$. Let $\eta(\mathbf{B})$ denote the maximum number of congestion constraints (in MIP-(7.14b)) in which a path in $\mathcal{F}'$ is simultaneously involved; the definition is based on the fact that a path f appears in every congestion constraint corresponding to the links on f and those interfering with links on f. We need to minimize $\eta(\mathbf{B})$ to produces a "good" approximate ratio from the rounding scheme in later steps; however, it is possible that $\eta(\mathbf{B})$ may become $\Omega\left(\Delta_m \max_l |\mathcal{I}(l)|\right)$ if we use the original set of congestion constraints in LP-(7.13d). To control this, we coarsen the formulation with the following approach based on the our definition of the interference sets in the SINR model:

(1) For any link l, $\mathcal{I}(l)$ only includes the links in the same length class as l. We rewrite and relax the inequalities in MIP-(7.14b) in $g(L)$ rounds. In the ith round, we partition the plane into $\frac{d_i}{8} \times \frac{d_i}{8}$ grid cells. A cell is said to be *non-empty* if there is at least sender node of a link in L_i in it. Let $\mathcal{B}_i$ denote the set of non-empty cells and let b denote a cell in $\mathcal{B}_i$. We say $l \in b$ or b *contains* l if and only if l is in L_i and $x(l)$ is within a distance γd_i of any point in cell b. Since the path refinement step do not increase path lengths (in number of hops), all the path lengths are bounded by $2\Delta_m$. For each path $f \in \mathcal{F}'$, the number of cells in $\mathcal{B}_i$ that contain at least one link in $L(f)$ is $O\left(\Delta_m\right)$. Let $\mathcal{B}$ denote the union of all $\mathcal{B}_i$, for $i = 1, 2, \ldots, g(L)$.

(2) MIP-(7.14b) implies the following:

$$\forall b \in \mathcal{B}, \sum_{l \in b} \sum_{f \in \mathcal{F}' : l' \in L(f)} z(f) \leq w. \tag{7.15}$$

We scale the coefficients of any rate variable $z(f)$ in Inequality (7.15) down by a constant factor of K_0 (which upper-bounds the number of links of a path that lie in the same interference set (as a result of Step 3.3)), such that the coefficients in the inequality above falls in $[0,1]$. That renders a relaxed set of congestion constraints as

$$\forall b \in \mathcal{B},\ \frac{1}{K_0} \sum_{l \in b} \sum_{f \in \mathcal{F}' : l \in L(f)} z(f) \leq w.$$

This can be rewritten in an aggregate manner as $\mathbf{B}'\vec{z} \leq \vec{w}$, where $\mathbf{B}'$ is a $[0,1]^{|\mathcal{B}| \times |\mathcal{F}'|}$ matrix (with $|\mathcal{B}| \leq |\mathcal{V}|$ as the number of non-empty cells). After the relaxation, $\eta(\mathbf{B}') = O(\Delta_m)$.

(Step 3.5) **Pre-processing: MIP reformulation**. We reformulate (MIP) into (MIP-1) as below. The constant parameter γ associated with the interference sets defined in the SINR setting enables us to adapt solutions to (MIP-1) to (MIP).

$$\begin{aligned}
\textbf{MIP-1:} \quad & \min \quad w \\
& \text{s.t.} \quad \sum_{f \in \mathcal{F}'_i} x(f) = 1,\ \forall i = 1, \ldots, k && (7.16a) \\
& \qquad \mathbf{B}'\vec{x} \leq \vec{w} && (7.16b) \\
& \qquad x(f) \in \{0,1\},\ \forall f \in \mathcal{F}' && (7.16c)
\end{aligned}$$

Since any interference set touches at most $O(\gamma^2)$ cells, any solution to the (MIP-1) with an objective value w is a solution to (MIP) and produces an objective value for (MIP) which is $O(\gamma^2 K_0)w = O(w)$, where γ and K_0 are constant.

(Step 3.6) **Rounding process**. We reuse the solution technique for Step 3.6 in Section 6.6 to solve (MIP-1) and obtain the following lemma. Notice that we have a factor $g(L)$ in Lemma 7.16-(3) carried on from the previous steps.

Lemma 7.16. *Let $\mathcal{F}'' = \{f : x'(f) > 0\}$ denote the set of selected paths with positive rates; $\mathcal{F}'' \subseteq \mathcal{F}'$. The rounding process ensures that (1) for each $f \in \mathcal{F}''$, $x'(f) = 1$; (2) for each connection c, $x'(c) = \sum_{f \in \mathcal{F}''(c)} x'(f)$. (3) $|\mathcal{F}''| = \sum_c x'(c) = \Theta(\sum_c y^*(c)) = \Theta\big(g(L) \cdot OPT^{LP}(\Delta())\big)$; and (4) for each link l, we have $\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_c x'(l', c) \leq K_1 \log \Delta_m / \log\log \Delta_m$, where $x'(l', c)$ is the flow rate on path $f \in \mathcal{F}''(c)$ such that $l' \in L(f)$, and K_1 is a constant.*

(Step 3.7) **Post-processing: Scaling and choosing flow vector**. We choose a rate vector $\lambda()$ as $\lambda(f) = K_2 \frac{\epsilon' - \epsilon}{e\Gamma g(L)} \frac{\log\log \Delta_m}{\log \Delta_m} x'(f)$, $\forall f \in \mathcal{F}''$, where K_2 is a constant, such that the congestion constraints in LP-(7.13d) are satisfied. Note that for some connections, multiple flows might be chosen (whereas for some connections, none would be chosen); for each ("original") connection c, define $\lambda(c) = \sum_{f \in \mathcal{F}''(c)} \lambda(f)$ as the total flow of c. $\sum_c \lambda(c) = \frac{\log\log\Delta_m}{\log\Delta_m}\Omega\big(OPT^{LP}(\Delta())\big)$. As discussed before, $OPT^{LP}(\Delta()) = \Omega\big(OPT(\Delta())/g(L)\big)$. Now Lemma 7.15 follows, and our algorithm ends.

Combining the delay analysis in Section 7.3.3, gives us Theorem 7.13 and Corollary 7.14 — the bi-criteria approximation — for the SU-DCTM problem.

7.5 Chapter Summary and Discussion

In this chapter, we extend our problem scope to cognitive networks and physical interference, and develop an optimization framework — a cross-layer solution for maximizing throughput for secondary users with delay requirements while maintaining throughput rates for existing primary users. Our optimization framework allows the computation of throughput-delay tradeoffs for arbitrary networks. The basis of our cross-layer optimization solution is a rate-based distributed scheduling and power control policy RB-Sched-SINR with worst-case analytical bounds for both the throughput region and delays; this is the first low-complexity scheduling algorithm that provides end-to-end delay upper-bounds which are independent of network size. The throughput region of RB-Sched-SINR is reasonably good: in many cases it can be order optimal (because $g(L)$ is small in practice). Direct extensions of this work include multi-channel multi-radio networks and channel switching delays. Another significant and natural avenue for future work is to extend our work to asynchronous settings (*e.g.*, with 802.11 protocol) under the physical interference model and explore the back-off and collision avoidance schemes, thus both making theoretical and practical contributions.

Chapter 8

System Model for Distributed Computing in Chapters 9, 10 and 11

Table 8.1: Notation for distributed computing in Chapters 9, 10 and 11.

G	network graph	$d(u,v)$	dist. of u and v
V	set of nodes	L	set of links
n	#nodes	$g(L)$	link diversity
m	#links	$OPT()$	optimum instance
α	path-loss exponent	$x(l)$	sender of link l
β	SINR threshold	$r(l)$	receiver of link l
N	background noise	$d(l)$	length of link l
A	affectance	SP	sensed power
D	network diameter	μ	distance diversity

8.1 Network Model

The wireless network is modeled as a directed graph $G = (V, L)$ in a Euclidean plane.We let V denote a set of tranceivers (henceforth, referred to as nodes) in the Euclidean plane. We assume L is a set of links with end-points in V, which form the set of communication requests for the maximum link scheduling problem, and $|L| = m$. Links are directed, and for link $l = \big(x(l), r(l)\big)$, $x(l)$ and $r(l)$ denote the transmitter (or sender) and receiver respectively. For a link set L', let $X(L')$ denote the set of senders of links in L', and likewise $R(L')$ the set of receivers. Let $d(u, v)$ denote the Euclidean distance between nodes u, v. For link l, let $d(l) = d(x(l), r(l))$ denote its link length. For links l, l', let $d(l', l) = d(x(l'), r(l))$. Let d_{min} and d_{max} denote the smallest and the largest transmission link lengths respectively. Let $B(v, d)$ denote the ball centered at node v with a radius of d. Each sender $x(l)$ uses power $P(l)$ for transmission on l; we assume commonly used path loss models [25, 46], where a transmission on link l is possible only if:

$$\frac{\frac{P(l)}{d^{\alpha}(l)}}{N(1+\phi)} \geq \beta, \tag{8.1}$$

where $\alpha > 2$ is the "path-loss exponent", $\beta > 1$ is the minimum SINR required for successful reception, N is the background noise, and $\phi > 0$ is a constant (note that α, β, ϕ and N are all constants).

We partition the set of transmission links into non-overlapping link classes. We define *link diversity* $g(L) = \lceil \log_2 \frac{d_{max}}{d_{min}} \rceil$. Partition $L = \{L_i\}, i = 1, 2, \ldots, g(L)$, where each $L_i = \{l \mid 2^{i-1} d_{min} \leq d(l) < 2^i d_{min}\}$ is the set of links of roughly similar lengths. Let $d_i = 2^i d_{min}$, such that d_i is an upperbound of link lengths in L_i; and $\forall i, \forall l \in L_i$, we define $\hat{d}(l) = d_i$. In a distributed environment, nodes use their shared estimates of minimum and maximum possible link length to replace d_{min} and d_{max}. $g(L) \leq \log 10^6 < 20$ in most cases*. further, as discussed earlier, each link can compute which link class it belongs to. The *reverse link* of a link l, denoted by $\overleftarrow{l}$, is the same link with transmission direction inverted. For a link set L', We use $\overleftarrow{L'}$ to denote the set of reverse links of L'.

8.2 Wireless Interference

We use physical interference model based on geometric SINR constraints (henceforth referred to as the SINR model), where a subset $L' \subseteq L$ of links can make successful transmission

*d_{min} is constrained by device dimension, empirically at least 0.1 meter; d_{max} depends on the type of the network, and is usually bounded by 10^5 meters. For example, in Bluetooth, 60GHz or Wi-Fi networks the typical transmission ranges are below 100 meters, and even experimental long-distance Wi-Fi networks have a transmission limit of 100 kilometers.

simultaneously if and only if the following condition holds for each link $l \in L'$:

$$\frac{\frac{P(l)}{d^{\alpha}(l)}}{\sum_{l' \in L' \setminus \{l\}} \frac{P(l')}{d^{\alpha}(l',l)} + N} \geq \beta. \tag{8.2}$$

Such a set L' is said to be *independent* in the context.

8.3 Distributed Computing Model in the SINR-based Model

Traditionally, distributed algorithms for wireless networks have been studied in the radio broadcast model [20,100,126] and its variants. The SINR based computing model is relatively recent, and has not been studied that extensively. Therefore, we summarize the main aspects and assumptions underlying this model: (1) The network is synchronized and for simplicity we assume all slots have the same length. (2) All nodes have a common estimate of m, the number of links, within a polynomial factor; (3) For each link $l \in L$, $x(l)$ and $r(l)$ have an estimate of $d(l)$, but they do not need to know the coordinates or the direction in which the link is oriented; (4) All nodes share a common estimate of d_{min} and d_{max}, the minimum and maximum possible link lengths; (5) We make critical use of RSSI measurement from carrier sensing in our algorithm as a way of communication such that nodes gain information by only "listening" the channel when they can instead of the need of understanding messages. As discussed in [125], given a threshold $Thres$, nodes are able to detect if the sensed power strength is $\geq Thres$.

8.4 Sensed Power-strength and Affectance

For ease of analysis based on links, we define *affectance* as that in [53, 55]: the affectance of link l caused by link l' is defined as $A(l', l) = c_l \dfrac{P(l')/d^{\alpha}(x(l'), r(l))}{P(l)/d^{\alpha}(l)}$, where $c_l = \beta/(1 - \beta N d^{\alpha}(l)/P(l))$. For a set L' of links, this is extended to $A(L', l) = \sum_{l' \in L'} A(l', l)$ and $A(l, L') = \sum_{l' \in L'} A(l, l')$. It can be verified that Inequality (8.2) is equivalent to $A(L' \setminus \{l\}, l) \leq 1$, signifying the success of data transmission on l. Following [53, 55], we say that a set L' of links is a δ-signal set if and only if $A(L' - \{l\}, l) \leq 1/\delta$ for each $l \in L'$.

To simplify the analysis based on nodes, we define *sensed power-strength* $SP(w, v)$, as the signal power that node v receives when only w is transmitting (which includes background noise); that is, $SP(w, v) = P(w)/d^{\alpha}(w, v) + N$. Likewise, we have SP from a node set W: $SP(W, v) = \sum_{w \in W} P(w)/d^{\alpha}(w, v) + N$.

8.5 Node Capabilities for Distributed Scheduling

Wireless node capabilities, such as half/full duplex communication and non-adaptive/adaptive power, may play a vital role in the distributed computation and scheduling.

8.5.1 Half/full Duplex Communication

Wireless radios are generally considered *half duplex*, *i.e.*, with a single radio they can either transmit or receive/sense but not both at the same time. *Full duplex* radios, which are becoming reality [60, 76] (through some self-interference canceling design), enable devices to transmit and receive simultaneously in the same frequency.

8.5.2 Non-adaptive/adaptive Power

By "adaptive power control" we mean that nodes can transmit at any power level in the range $[0, P_{max}]$. The fact that all the links in MaxLSP-U use a uniform power level for data transmission, does not necessarily constrain their capability of employing adaptive-power radios which can vary across different transmission power levels to be *used for scheduling or control*.

8.6 Sub-linear Length-monotone Power Assignment

A power assignment is *sub-linear length-monotone* if $\frac{d(l)}{d(l')} \leq \frac{P(l)}{P(l')} \leq \frac{d^{\alpha}(l)}{d^{\alpha}(l')}$ whenever $d(l) \geq d(l')$. Intuitively, that is to say, the change (increase or decrease) rate of the transmission power on a link do not exceed the change rate of the signal fading due to spatial separation. One property of sub-linear length-monotone power assignments is that if $d(l)$ and $d(l')$ are within a constant factor of each other, then the ratio between $P(l)$ and $P(l')$ is (upper- and lower-) bounded by constants.

Chapter 9

Fast Distributed Algorithms for Uniform-power MIS and One-shot Local Broadcasting under the Physical Interference Model

9.1 Preliminaries and Definitions

In this chapter, we consider the uniform-power version of MaxLSP and MaxLBP where all the nodes in the solution set are required to use the same power level for transmission, thus simplifying the expression for SINR, sensed power-strength and affectance; and we use P to denote the value of the uniform data transmission power, based on which we have the following concept of threshold for sensed power-strength. Note that nodes have freedom to choose the transmission power when computing the solution. Let $Thres(d) = P/d^{\alpha} + N$ be a function of distance d, such that for a node v, if any other node is transmitting in a range of d, its sensed power will exceed $Thres(d)$. Note that the use of uniform data transmission power does not limit us to use the same transmission power for control/scheduling purpose; and we provide solutions to accommodate both the cases with and without the constraint of uniform transmission power for control/scheduling.

Analogously to the link diversity $g(L)$, we define range diversity $g(V)$ of a set V of nodes in MaxLBP, as the $g(V) = \lceil \log \frac{\max_{v \in V} d(v)}{\min_{v \in V} d(v)} \rceil$, where $d(v)$ denotes the requested local broadcast range for node v.

9.2 Problem Definition

9.2.1 The Maximum Link Scheduling Problem for Uniform-power Data Transmission (MaxLSP-U)

Given a set of communication requests (links) L, the goal of the MaxLSP problem is to find a maximum independent subset of links that can be scheduled simultaneously in the SINR model. MaxLSP-U is an instance of MaxLSP where links in a solution use a uniform power level for data transmission; note that this does not necessarily restrict scheduling to uniform power. In this chapter, we use $OPT(L)$ to denote an optimum solution to the MaxLSP-U, and thus $|OPT(L)|$ is the cardinality of the largest such independent set. As discussed earlier, computing $OPT(L)$ is NP-hard, and we focus on approximation algorithms. We say an algorithm gives a C-approximation factor if it constructs an independent link set $L' \subseteq L$ with $|L'| \geq |OPT(L)|/C$.

9.2.2 The One-shot Local Broadcasting Problem (MaxLBP)

Given a set V of broadcast nodes, and a range $d(v)$ for each node $v \in V$, the goal is to select a subset $V' \subset V$, so that when all the nodes in V' broadcast simultaneously, for each node $v \in V'$, any potential receiver within distance $d(v)$ from v can receive v's broadcast message successfully. That is, we need to keep the SINR above β for any location within the range of each selected broadcast node in V'. In this problem, we assume each node uses a uniform power level $P(v) = P$. We use $OPT_{\text{MaxLBP}}(V)$ to denote the optimal solution for the instance of MaxLBP; we drop the subscript whenever it is clear from the context. Here again, we study approximation algorithms, and the goal is to approximate $OPT_{\text{MaxLBP}}(V)$.

9.3 Distributed Algorithm: Overview

In this section, we present the distributed algorithm for MaxLSP-U. Because the algorithm is quite complicated, we briefly summarize the sequential algorithm of [44, 55, 137] below, and then give a high-level description of the distributed algorithm and its analysis, without the implementation details of the individual steps in the SINR model. Section 9.4 describes the algorithm for computing a ruling in the full and half duplex models. The complete distributed implementation and other details are discussed in Sections 9.5 and 9.6, for the non-adaptive and adaptive power control settings, respectively.

9.3.1 The Centralized Algorithm

We discuss the centralized algorithm adapted from [44, 55, 137] for MaxLSP-U, which forms the basis for our distributed algorithm. The algorithm processes links in non-decreasing order of length. Let L be the initial set of links, and S the set of links already chosen (which is empty initially). Each iteration involves the following steps:

(1) picking the shortest link l in $L \setminus S$ and removing l from L,

(2) removing from L all the links in $\{l' \in L \setminus S : A(S, l') \geq c_0\}$ where $c_0 < 1$ is a constant, *i.e.*, all the links in $L \setminus S$ that suffer from high interference caused by all chosen links in S, and

(3) removing from L all the links in $\{l' \in L \setminus S : d(l', l) = c_1 d(l)\}$ where c_1 is a constant, *i.e.*, all the nearby links of l in $L \setminus S$.

The results of [44, 55, 137] show that: S is feasible (*i.e.*, the SINR constraints are satisfied at every link), and $|S|$ is within a constant factor of the optimum. Consider a link l that is added to S in iteration i. The proof of feasibility of set S involves showing that for this link l, the affectance due to the links added to S after iteration i is at most $1 - c_0$, so that simultaneous transmission by all the links in S does not cause high interference for l. The approximation factor involves the following two ideas: (1) for any link $l \in S$, there can be at most $O(1)$ links in $OPT(L)$ which are within distance $c_0 d(l)$, and (2) in the set of links removed in step 2 due to the affectance from S, there can be at most $O(1)$ links in $OPT(L)$. We note that the second and third steps are reversed in [137], while [55] does not use the third step. However, we find it necessary for our distributed algorithm, which uses the natural approach of considering all the links in a given length class simultaneously (instead of sequentially). Our analysis builds on these ideas, and property (1) holds for our case without any changes. However, property (2) is more challenging to analyze, since many links are added in parallel. Another complication is that the distributed implementation has to be done from the senders' perspective, so that the above steps become more involved.

9.3.2 Additional Definitions

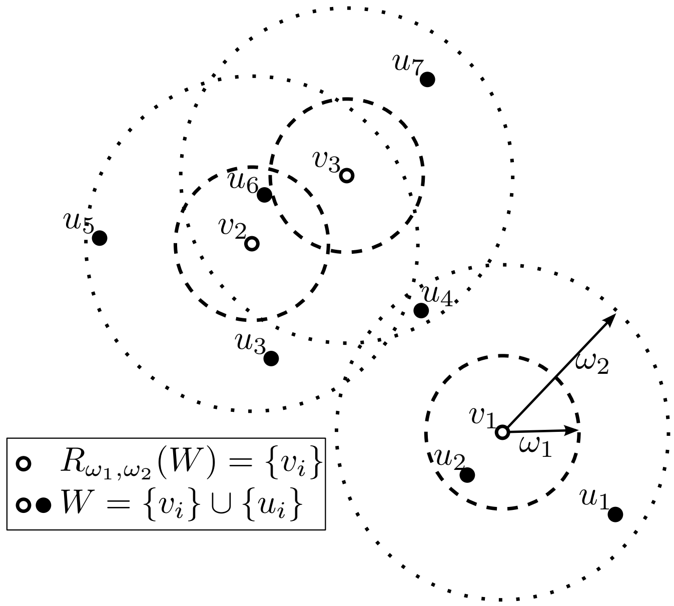


Figure 9.1: Example of an (ω_1, ω_2)-ruling: $W = \{v_i\} \cup \{u_i\}$ is the set of all dots (open and dark), while $R_{\omega_1,\omega_2}(W) = \{v_i\}$ which is the set of all the open dots denotes a (ω_1, ω_2)-ruling of W. Note that all the nodes in W are ω_2-covered by $R_{\omega_1,\omega_2}(W)$, while all the nodes in $R_{\omega_1,\omega_2}(W)$ are ω_1 away from each other.

Cover and Ruling. Let W, W' denote two node sets. We say a node u is ω-*covered* by W', if and only if $\exists u' \in W', d(u, u') \leq \omega$; based on that, we say W is ω-*covered* by W', or equivalently W' ω-*covers* W, if and only if every node in W is ω-covered by W'. An (ω_1, ω_2)-*ruling* (where $\omega_1 < \omega_2$) of W, introduced in [31], is a node set denoted by $R_{\omega_1,\omega_2}(W)$, such that

(1) $R_{\omega_1,\omega_2}(W) \subseteq W$;
(2) all the nodes in $R_{\omega_1,\omega_2}(W)$ are at least ω_1-separated; that is, $\forall u, u' \in R_{\omega_1,\omega_2}(W)$, $d(u, u') \geq \omega_1$; and
(3) W is ω_2-covered by $R_{\omega_1,\omega_2}(W)$.

Here, we have adopted a generalized definition by considering Euclidean distance rather than graph distance. The concept of ruling has a vital role in our algorithm: it is used for choosing a set of spatially separated links and removing the nearby links of the chosen links. Figure 9.1 gives an example to illustrate these notions.

Algorithm 2: Distributed Maximum Link Scheduling

input : Set L of links

output: One-shot Schedule S

1 $J \leftarrow L$;

2 **foreach** $i = 1, 2, \ldots, g(L)$ **do** /* for each link class in L */

3 $J_i \leftarrow L_i \cap J$, $J_i^{>} \leftarrow \cup_{j>i} L_j \cap J$, $\omega_1 \leftarrow \gamma_1 d_i, \omega_2 \leftarrow \gamma_2 d_i$;

/* 1st step (Lines 4-6): check affectance constraints, s.t. links in S_i are not subject to high interference by links of smaller lengths chosen in previous phases. Note: we use reverse l to check affectance at $x(l)$ */

4 **if** $i > 1$ **then**

5 $J_i^a \leftarrow \left\{ l \in J_i : A(\cup_{j<i} S_j, \overleftarrow{l}) \le \psi(1 - (\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha \right\}$, $\overline{J_i^a} \leftarrow J_i \setminus J_i^a$;

6 $J_i^b \leftarrow \left\{ l \in J_i^{>} : A(\cup_{j<i} S_j, \overleftarrow{l}) \le \psi(1 - (\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha \right\}$, $\overline{J_i^b} \leftarrow J_i^{>} \setminus J_i^b$;

/* 2nd step (Lines 7 & 8): check spatial constraints, to obtain an (ω_1, ω_2)-ruling $X(J_i^r)$ of $X(J_i^a)$, s.t. the selected links (J_i^r) are spatially separated and nearby links (J_i^z) of similar or larger lengths are excluded */

7 construct link set J_i^r, s.t. $X(J_i^r)$ is an (ω_1, ω_2)-ruling of $X(J_i^a)$;

8 construct link set J_i^z, s.t. (1) $J_i^z \cap J_i^a = J_i^a \setminus J_i^r$ and (2) $\{l \in J_i^b :$ $x(l)$ is ω_1-covered by $X(J_i^r)\} \subseteq J_i^z \cap J_i^b \subseteq \{l \in J_i^b : x(l)$ is ω_2-covered by $X(J_i^r)\}$;

/* 3rd step (Lines 9 & 10): select and discard links */

9 $S_i \leftarrow J_i^r$, $J \leftarrow J \setminus J_i^r$; /* select links */

10 $J \leftarrow J \setminus (\overline{J_i^a} \cup \overline{J_i^b} \cup J_i^z)$; /* discard links */

11 $S \leftarrow \cup_i S_i$, Return S;

9.3.3 High-level Description of the Distributed Algorithm

We have provided a detailed discussion of the centralized algorithms and basic ideas in Section 9.3.1; now we discuss the distributed algorithm at a high level (Algorithm 2), and prove the main properties. We use the following constants in the algorithm $\gamma_1 = \left(\frac{36\beta}{1-\psi}\frac{\alpha-1}{(\alpha-2)}\frac{1+\phi}{\phi}\right)^{1/\alpha} + 2$ and γ_2 as an arbitrary constant $> \gamma_1$, where α, β, ϕ are constants in described in Inequality (8.1) and ψ is a constant that can take any value from $(0, 1)$. The algorithm sweeps through the link classes in $g(L)$ phases. In the ith phase, where $i \in [1, g(L)]$, it selects a subset of links from L_i to include in S, and removes a subset of links from $\cup_{j>i} L_j$ to speed up later phases; the comments in Algorithm 2 explain each step. Step 1 in line 5-8 of the algorithm eliminate links which do not satisfy affectance constraints — its implementation is formally described in Sections 9.5 and 9.6, and crucially relies on the measurement of the received power at each sender. Step 2 in line 9 of the algorithm constructs the distributed ruling, which is discussed in Section 9.4 for the non-adaptive power case, and extended to the adaptive case in Section 9.6.

Lemma 9.1 (Correctness)**.** *The algorithm results in an independent set S.*

Lemma 9.1 shows the correctness and is proved in Appendix 9.9.1. Algorithm 2 results in a constant approximation ratio shown in Theorem 9.2; its proof is in Appendix 9.9.2, where this theorem is backed by Lemma 9.14 and 9.15 in Appendix 9.9.2. The two lemmas, independent of Algorithm 2, stands on its own to provide insights of how an optimum solution is shaped under the SINR model; they can be proved by using a combination of techniques found in [55, 137]. We provide their proofs in Appendix 9.9.2 to make the chapter complete and help the reading.

Theorem 9.2 (Approximation Ratio)**.** *$\forall\gamma_1, \gamma_2, \psi > 0$, $|OPT(L)| \leq C_3(\gamma_2, \psi)|S|$ if $\gamma_2 > \gamma_1 > 1$, where $C_3(\gamma_2, \psi) = C_1(\gamma_2) + C_2(\psi(1 - (\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha)$, where $C_1(x) = \frac{(2x+1)^\alpha}{\beta}$ and $C_2(x) = (\frac{2(\beta b)^{1/\alpha}}{(\beta b)^{1/\alpha}-1})^\alpha/x + 1$. are functions with constant output values for constant input arguments.*

9.4 Distributed Algorithm: (ω_1, ω_2)-Ruling

In this section, we present Algorithm 3, the distributed algorithm to compute an (ω_1, ω_2)-ruling, for full duplex communication under the physical interference model; in the end of the section, we extend it to the half duplex setting (where a node can perform transmission and reception/sensing at the same time) with added running time. While this algorithm can be interesting by itself, it serves as a significant building block for our distributed implementation. For the algorithm to function properly, we require the input parameter $\omega_2 \geq (36\frac{\alpha-1}{\alpha-2})^{\alpha-2}\omega_1$. Recall that $B(v, d)$ denotes the ball centered at v with a radius of d. Let n be the total number of nodes. The last input parameter b_{max} denote the estimate of the maximum number of nodes in the ball $B(v, \omega_1)$ of any node $v \in W_1$; in the worst case, $b_{max} \leq n$.

In this algorithm, we call an iteration of the outer loop (Line 1) a *phase*; we call an iteration of the inner loop (Line 2) a *round*, consisting of the *coordination step* (Lines 4 through 8) and the *decision step* (Lines 9 through 12). A node v is said to be *active* if v has not joined either $\hat{R}$ or $\hat{Z}$; otherwise, v becomes *inactive*.

Algorithm 3: ConstructR$(\omega_1, \omega_2, W_1, W_2, b_{max})$: Distributed algorithm for computing an (ω_1, ω_2)-ruling with full duplex radios.

input : $\omega_1, \omega_2, W_1, W_2, b_{max}$
output: $(\hat{R}, \hat{Z})$: an (ω_1, ω_2)-ruling of W_1

`/* Each` $v \in W_1 \cup W_2$ `does the following */`
1 **for** $i_{out} = 0$ *to* $\log b_{max} + 1$ **do**
2   **for** $i_{in} = 1$ *to* $C_4 \log n$ **do**
3   **if** v *is active* **then**
            `/* Coordination Step (Lines 4-8): 1 slot */`
4             $U(v) \leftarrow 0$;
5   **if** $v \in W_1$ **then** $U(v)$ flips to 1 w/ prob. $\frac{2^{i_{out}-2}}{b_{max}}$;
6   **if** $U(v) = 1$ **then**
7   v transmits and senses, $I(v) \leftarrow$ the power v receives in this slot;
8   **if** $I(v) > Thres(\omega_1)$ **then** $U(v) \leftarrow 0$;
            `/* Decision Step (Lines 9-12): 1 slot */`
9   **if** $U(v) = 1$ **then** v transmits, v joins $\hat{R}$; `/* inactive */`
10   **else**
11   v senses, $I(v) \leftarrow$ the power v receives in this slot;
12   **if** $I(v) > Thres(\omega_1)$ **then** v joins $\hat{Z}$; `/* inactive */`
13 **return** $(\hat{R}, \hat{Z})$;

In each round, the coordination step provides a probabilistic mechanism for active nodes to compete to get in the ruling (at Line 5). Lines 6 through 8 constitute a module to resolve the issue of sensing and transmitting at the same time, such that two nearby nodes do not both enter the ruling (*i.e.*, Lemma 9.5). Next, during the decision step, a subset of active nodes decide to join $\hat{R}$ or $\hat{Z}$.

In each phase, there are $C_4 \log n$ rounds, such that we can ensure a fraction of the node population have either joined $\hat{R}$ or $\hat{Z}$, and we expect the maximum number of active nodes in the nearby region of any active node to decrease by a half (proved in Lemma 9.18 in Appendix 9.10.1). After each phase, the probability for each active node to access the channel and compete doubles (at Line 5). After the total of $\log b_{max} + 2$ phases, we have Lemmas 9.4, 9.6, 9.7 that lead to Theorem 9.3.

Theorem 9.3 (Correctness)**.** *Algorithm 3 terminates in $O(\log n \log b_{max})$ time. By the end of the algorithm: (1) $\hat{R}$ forms an (ω_1, ω_2)-ruling of W_1, and (2) $\hat{Z} \cap W_1 = W_1 \setminus \hat{R}$ and $\{v \in W_2 : v$ is ω_1-covered by $\hat{R}\} \subseteq \hat{Z} \cap W_2 \subseteq \{v \in W_2 : v$ is ω_2-covered by $\hat{R}\}$, w.h.p.*

Theorem 9.3 follows directly from the lemmas below. Lemmas 9.4, 9.5 and 9.7 prove that $\hat{R}$ is an (ω_1, ω_2)-ruling of W_1, w.h.p. Lemmas 9.4, 9.6, 9.7 together shows that $\hat{Z}$ complements $\hat{R}$ in W_1 and partially in W_2 with desired properties, w.h.p. To help the reading flow, we defer much of the technical content — the proof of Lemma 9.4 (which involves Lemmas 9.16, 9.17 and 9.18) and the proof of Lemma 9.7 — to Appendix 9.10.

Lemma 9.4 (Completion)**.** *By the end of the algorithm, all nodes in W_1 have joined either $\hat{R}$ or $\hat{Z}$,* i.e.*, all nodes in W_1 become inactive, w.h.p.*

Lemma 9.4 implies that $\hat{Z} \cap W_1 = W_1 \setminus \hat{R}$. We say a node $v \in \hat{R}$ is "*good*," if and only if $d(v, v') \geq \omega_1, \forall v' \in \hat{R}$ and $v' \neq v$. In Algorithm 3, When a node enters $\hat{R}$, it makes sure that there are no other ones entering $\hat{R}$ within a range of ω_1, and it deactivate all the active nodes in the same range. Therefore, we have the following Lemmas 9.5 and 9.6.

Lemma 9.5 (Quality of $\hat{R}$)**.** *All nodes in $\hat{R}$ are good, with probability of 1.*

Lemma 9.6 (Quality of $\hat{Z}$: Part 1)**.** *$\hat{Z}$ contains all the nodes in $W_1 \cup W_2 \setminus \hat{R}$ that are ω_1-covered by $\hat{R}$, with probability of 1.*

Lemma 9.7 (Quality of $\hat{Z}$: Part 2)**.** *Further, suppose all nodes in $\hat{R}$ are good, then all nodes in $\hat{Z}$ are ω_2-covered by $\hat{R}$, $\forall \omega_2 \geq (36\frac{\alpha-1}{\alpha-2})^{\frac{1}{\alpha-2}}\omega_1$.*

Half Duplex Communication. Now, we assume that nodes are in the half duplex mode, so that they cannot perform transmission and reception/sensing at the same time. In Algorithm 3, Lines 6 through 8 make use of the full duplex capability, such that Lemma 9.5 is true. To account for the case of half duplex communication, if we replace the one-slot deterministic full duplex mechanism (Lines 6 through 8) with a randomized $O(\log n)$-time loop — illustrated by the following lines of pseudo code — we have Lemma 9.8 for half duplex communication as the counterpart of Lemma 9.5 for full duplex. The cost incurred includes (1) the increase in the total running time to obtain an (ω_1, ω_2)-ruling by $O(\log n)$, and (2) a weakened statement in Lemma 9.8 compared to Lemma 9.5.

Lemma 9.8 (Quality of $\hat{R}$: Half Duplex Mode)**.** *All nodes in $\hat{R}$ are good, w.h.p.*

Since Lemmas 9.4, 9.6 and 9.7 remain valid, we obtain the following theorem for the half duplex case.

In replacement of Lines 6 through 8 in Algorithm 3 for using half duplex radios.

for $j = 1$ *to* $C_5 \log n$ **do** `/* resolving half duplex communication */`
    `/* in each slot */`
    if $U(v) = 1$ **then** v transmits with prob. $1/2$;
    if v *does not transmit* **then**
        v senses, $I(v) \leftarrow$ the power v receives in this slot;
        if $I(v) > Thres(\omega_1)$ **then** $U(v) \leftarrow 0$; `/* stops */`

Theorem 9.9 (Half Duplex)**.** *There exists a modified version of* ConstructR$(\omega_1, \omega_2, W_1, W_2, b_{max})$ *for the half duplex case, such that it finishes in* $O(\log^2 m \log b_{max})$ *time and by the end of the algorithm: (1)* $\hat{R}$ *forms an* (ω_1, ω_2)*-ruling of* W_1*, and (2)* $\hat{Z} \cap W_1 = W_1 \setminus \hat{R}$ *and* $\{v \in W_2 : v$ *is* ω_1*-covered by* $\hat{R}\} \subseteq \hat{Z} \cap W_2 \subseteq \{v \in W_2 : v$ *is* ω_2*-covered by* $\hat{R}\}$*, w.h.p.*

9.5 Distributed Implementation with Non-adaptive Uniform Transmission Power for Scheduling

Putting everything together, we present in this section the distributed implementation of Algorithm 2 when restricted to using one uniform power level for scheduling.

Theorem 9.10 (Performance)**.** *Our distributed implementation of Algorithm 2 with non-adaptive uniform transmission power has the following properties:*

(1) in half duplex mode, it terminates in $O(g(L) \log^3 m)$ *time,*
(2) in full duplex mode, it terminates in $O(g(L) \log^2 m)$ *time, and*
(3) in both modes, it produces a constant-approximate solution to MaxLSP-U*.*

For the ith phase of Algorithm 2, we present the distributed implementation that works even when there is only one fixed power level available. We assign γ_2 a constant value $\geq (36\frac{\alpha-1}{\alpha-2})^{\alpha-2}\gamma_1$, and let $\omega_1 = \gamma_1 d_i$, and $\omega_2 = \gamma_2 d_i$. The distributed implementation goes as follows.

Distributed Implementation: 1st Step: With Algorithm 5, we run CheckA$(J_i, J_i^>, \cup_{j<i} S_j)$ to implement the 1st step for phase i in Algorithm 2. $\forall l \in J_i \cup J_i^>$, on Line 3 of Algorithm 5,

Algorithm 5: CheckA(Y, Y', S): Distributed algorithm for checking affectance.

input : Link sets Y, Y', S

output: $Y^a = \{x(l) : l \in Y, A(S,l) \le \psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha\}$,
$Y^b = \{x(l) : l \in Y', A(S,l) \le \psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha\}$,
$\overline{Y^a} = \{x(l) : l \in Y, A(S,l) > \psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha\}$,
$\overline{Y^b} = \{x(l) : l \in Y', A(S,l) > \psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha\}$

`/* in 1 time slot: */`

if $l \in S$ **then** $x(l)$ transmits;
else if $l \in Y \cup Y'$ **then**
  $x(l)$ senses, $SP(X(S), x(l)) \leftarrow$ the power $x(l)$ receives;
  if $SP(X(S), x(l)) \le Thres\big((\frac{\beta(1-\frac{d(l)}{P/(\beta N)})}{\psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha})^{1/\alpha} d(l)\big)$ **then**
    if $l \in Y$ **then** $x(l)$ joins Y^a; **if** $l \in Y'$ **then** $x(l)$ joins Y^b;
  else
    if $l \in Y$ **then** $x(l)$ joins $\overline{Y^a}$; **if** $l \in Y'$ **then** $x(l)$ joins $\overline{Y^b}$;
return $Y^a, Y^b, \overline{Y^a}, \overline{Y^b}$;

we get $SP(X(S), x(l)) = \sum_{l' \in \cup_{j<i} S_j} \frac{P}{d^\alpha(l', \overleftarrow{l})} + N$. Then, since

$$SP(X(S), x(l)) \le Thres\big((\frac{\beta(1-\frac{d(l)}{P/(\beta N)})}{\psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha})^{1/\alpha} d(l)\big)$$

is equivalent to

$$A(\cup_{j<i} S_j, \overleftarrow{l}) = \frac{\beta}{1-\frac{d^\alpha(l)}{P/(\beta N)}} \sum_{l' \in \cup_{j<i} S_j} \frac{d^\alpha(l)}{d^\alpha(l', \overleftarrow{l})} \le \psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^\alpha,$$

the sets of links whose sender nodes are in $Y^a, Y^b, \overline{Y^a}, \overline{Y^b}$ correspond to $J_i^a, J_i^b, \overline{J_i^a}, \overline{J_i^b}$ in Algorithm 2 respectively.

Distributed Implementation: 2nd Step: Recall that for a link set L', $X(L')$ is the set of all sender nodes. To implement the 2nd step for phase i in Algorithm 2. we feed $b_{max} = m$ to Algorithm 3 and run ConstructR$(\omega_1, \omega_2, X(J_i^a), X(J_i^b), m)$. Thus, we obtain an (ω_1, ω_2)-ruling $\hat{R}$ of $X(J_i^a)$ and $\hat{Z}$ that complements $\hat{R}$ in $O(\log^3 m)$ time for half duplex and $O(\log^2 m)$ time for full duplex, Then, the sets of links whose sender nodes are in $\hat{R}, \hat{Z}$ respectively correspond to $X(J_i^r), X(J_i^z)$ in Algorithm 2.

Distributed Implementation: 3rd Step: The 3rd step of Algorithm 2 means all the links in class L_i and those longer links removed in the 1st step exit Algorithm 2. Because our algorithm is sender based, the corresponding links will quit upon the decision of their sender nodes in the 1st and the 2nd steps.

9.6 Distributed Implementation with Adaptive Transmission Power for Scheduling

In this section, suppose we have multiple power levels at our disposal on each node*; we present how this aids the distributed implementation of Algorithm 2.

Theorem 9.11 (Performance)**.** *Our distributed implementation of Algorithm 2 with adaptive transmission power has the following properties:*

(1) in half duplex mode, it terminates in $O(g(L)\log^2 m)$ time,
(2) in full duplex mode, it terminates in $O(g(L)\log m)$ time, and
(3) in both modes, it produces a constant-approximate solution to MaxLSP-U*.*

Again, note that these adaptive power levels are only for scheduling in the control phase; for data transmission in the resulting independent set S we still use one uniform power level. Specifically, we require that (1) nodes have access to a set of $\Theta(g(L))$ power levels; and (2) for each $i \in [1, g(L)]$, there exists a power level P_i to use such that $(\frac{P_i}{\beta N})^{1/\alpha} = \gamma_3 d_i$, where γ_3 is a constant.

We present a second method to implement the 2nd step of each phase in Algorithm 2, reducing the running time by one logarithmic factor, by (1) performing a preprocessing to reduce b_{max} to some constant C_9 in $O(\log m)$ time, (2) running Algorithm 3 with the constant C_9 in $O(\log^2 m)$ time with half duplex radios and $O(\log m)$ time with full duplex radios, and (3) performing a postprocessing to obtain the sets of links required as a result of 2nd step of Algorithm 2 in one slot. We introduce a new constant $\gamma_4 \geq (36\frac{\alpha-1}{\alpha-2})^{\alpha-2}\gamma_1$, and assign γ_2 a constant value $\geq \gamma_3 + \gamma_4$. For the ith phase of Algorithm 2, let $\omega_1 = \gamma_1 d_i$, $\omega_2 = \gamma_2 d_i$, $\omega_3 = \gamma_3 d_i$ and $\omega_4 = \gamma_4 d_i$.

*In this chapter we only study MaxLSP-U where links in a solution use a uniform power level for data transmission; this does not necessarily restrict scheduling control to using uniform power. The general version of the problem MaxLSP that explores power control in both scheduling and data transmission in a distributed setting is a hard problem and remains open.

We reuse the implementation for the 1st and the 3rd steps from the previous section. We implement the 2nd step of each phase in Algorithm 2 with the following three sub-steps.

9.6.1 Preprocessing: Constant Density Dominating Set

Scheideler, Richa, and Santi in [125] propose a distributed protocol to construct a *constant density dominating set* of nodes under uniform power assignment within $O(\log m)$ slots. They define $Dom(W, P_t)$ as a dominating set of a node set W with transmission power of P_t on each node, such that W is d_t-covered by $Dom(W, P_t)$, where d_t is the transmission range under P_t. Then, by "constant density", they mean that $Dom(W, P_t)$ is a $O(1)$-approximation of the minimum dominating set of W, such that within the transmission range d_t of each node in W there are at most a constant number C_9 of nodes chosen by $Dom(W, P_t)$.

At phase i of Algorithm 2, after the 1st step of checking affectance, we execute the protocol on the node set $X(J_i^a)$ with power P_i which corresponds to a transmission range of ω_3, and thus obtain a constant density dominating set $Dom(X(J_i^a), P_i)$ out of $X(J_i^a)$. $Dom(X(J_i^a), P_i)$ has the following properties:

(1) $Dom(X(J_i^a), P_i) \in X(J_i^a)$;
(2) **dominating set**: all the node in $X(J_i^a)$ ω_3-covered by $Dom(X(J_i^a), P_i)$; and
(3) **constant density**: $\forall v \in X(J_i^a)$, $1 \leq |B(v, \omega_3) \cap Dom(X(J_i^a), P_i)| \leq C_9$, where C_9 is a constant.

9.6.2 Construction of Ruling $X(J_i^r)$

ConstructR$(\omega_1, \omega_4, Dom(X(J_i^a), P_i), X(J_i^b), C_9)$ produces $\hat{R}$ as an (ω_1, ω_4)-ruling of $Dom(X(J_i^a), P_i)$, and $\hat{Z}$ such that

(1) $\hat{Z} \cap X(J_i^a) = Dom(X(J_i^a), P_i) \setminus \hat{R}$;
(2) $\hat{Z} \cap X(J_i^b) \supseteq \{v \in X(J_i^b) : v \text{ is } \omega_1\text{-covered by } \hat{R}\}$; and
(3) $\hat{Z} \cap X(J_i^b) \subseteq \{v \in X(J_i^b) : v \text{ is } \omega_4\text{-covered by } \hat{R}\}$, *i.e.*, $\hat{Z}$ is ω_4-covered by $\hat{R}$.

We argue that $\hat{R}$ is an (ω_1, ω_2)-ruling of $X(J_i^a)$ due to the following two properties: (1) $\hat{R} \subseteq X(J_i^a)$ and any two nodes in $\hat{R}$ are ω_1-separated, and (2) $X(J_i^a)$ is ω_2-covered by $\hat{R}$. Property (2) can be deduced from the facts below:

(1) $X(J_i^a)$ is ω_3-covered by $Dom(X(J_i^a), P_i)$ due to the preprocessing step;
(2) $Dom(X(J_i^a), P_i)$ is ω_4-covered by $\hat{R}$ as a result of ConstructR$(\omega_1, \omega_4, Dom(X(J_i^a), P_i), X(J_i^b), C_9)$; and
(3) $\omega_4 + \omega_3 \leq \omega_2$ by our construction.

Therefore, $\hat{R}$ corresponds to $X(J_i^r)$ in the 2nd step of Algorithm 2.

9.6.3 Postprocessing: Accounting for $X(J_i^z)$

Construct $\hat{Z}' \triangleq \hat{Z} \cup \left(X(J_i^a) \setminus \hat{R}\right)$; the following is true:

(1) $\hat{Z}' \cap X(J_i^a) = X(J_i^a) \setminus \hat{R}$;
(2) $\hat{Z}' \cap X(J_i^b) \supseteq \{v \in X(J_i^b) : v \text{ is } \omega_1\text{-covered by } \hat{R}\}$; and
(3) $\hat{Z}' \cap X(J_i^b) \subseteq \{v \in X(J_i^b) : v \text{ is } \omega_2\text{-covered by } \hat{R}\}$, *i.e.*, $\hat{Z}'$ is ω_2-covered by $\hat{R}$.

Therefore, $\hat{Z}'$ corresponds to $X(J_i^z)$ in the 2nd step of Algorithm 2.

9.7 Distributed Algorithm for MaxLBP-U

MaxLBP-U is different from MaxLSP-U, since only the set V of senders is specified as the input, instead of sender-receiver pairs. Further, all potential receiving locations in range $d(v)$ for each $v \in V$ need to be considered. We only provide an algorithm under uniform power assignment, due to the challenge of no assistance from receiver nodes. We define the notion of affectance of node w on node v as: $A(w, v) = c_v \frac{d^\alpha(v)}{d^\alpha(w,v)}$, where $c_v = \frac{\beta}{(1-\beta N d^\alpha(v)/P)}$. It is extended to a set S of nodes as $A(S, v) = \sum_{w \in S} A(w, v)$. We say S is *feasible* if and only if S is a solution (not necessarily optimal) to MaxLBP-U. Note that there exists some constant $\psi' < 1$ such that S is feasible if $\forall v \in S, A(S \setminus \{v\}, v) \leq \psi'$. The sequential MaxLBP-U algorithm involves the steps below.

(1) Consider the nodes $v_1, \ldots, v_n$ in V in a sorted non-decreasing order of their ranges; $S = \emptyset$ initially;
(2) For $i = 1$ to n: if $A(S, v_i) < \psi'/4$, add v_i to S;
(3) Output $S' = \{v \in S : A(S \setminus \{v\}, v) \leq \psi'\}$.

Note that this algorithm returns a feasible solution by construction. The constant factor approximation guarantee follows by an adaptation of the Red-Blue Lemma of [55].

Algorithm 6: Distributed MAXLBP-U

input : Set V of nodes, with range $d(v)$ for each $v \in V$. We assume V_i is the set of nodes v with range $d(v) \in (2^{i-1}, 2^i]$. $d_i = 2^i$.
output: Subset V' of V for MAXLBP-U

$S'_i \leftarrow \emptyset,\ \forall i = 1 \ldots, g(V)$; `/* initialization */`
$S_i \leftarrow \emptyset, S'_i \leftarrow \emptyset, X_i \leftarrow \emptyset, R_i \leftarrow \emptyset,\ \forall i = 1 \ldots, g(L)$;
foreach $i = 1, 2, \ldots, g(V)$ **do** `/*` $g(V)$ `phases */`
  `/* check affectance */`
  if $v \in \cup_{j<i} S'_j$ **then**
    v transmits;
  if $v \in V_i$ **then**
    v senses, $SP_{<i}(v) \leftarrow$ the sensed power at v;
    $A^E(\cup_{j<i} S'_j, v) \leftarrow$ estimated $A(\cup_{j<i} S'_j, v)$;
    if $A^E(\cup_{j<i} S'_j, v) < \psi'/4$ **then** add v to S_i;
  $\omega_1 \leftarrow \gamma'_1 d_i, \omega_2 \leftarrow \gamma'_2 d_i$;
  $S'_i \leftarrow R_{\omega_1,\omega_2}(S_i)$;
foreach $i = g(V), \ldots, 2, 1$ **do** `/*` $g(V)$ `phases */`
  if $v \in S'_i$ **then**
    v transmits;
    if $\sum_j A^E(S'_j, v) < \psi'$ **then** add v to V';
  if $v \in \cup_{j<i} S'_j$ **then**
    v senses, $SP_i(v) \leftarrow$ the sensed power at v;
    $A^E(S'_i, v) \leftarrow$ estimated $A(S'_i, v)$;
return $\cup_i V'_i$

Algorithm 6 employs at Line 11 the Ruling construction subroutine, which outputs an (ω_1, ω_2)-ruling of S_i. Theorem 9.12 summarizes the performance of algorithm 6.

Theorem 9.12. *Algorithm 6 runs in time $O(g(V)\log^2 n)$ with full duplex radio, and produces a feasible solution for* MaxLBP-U *within a constant factor of $|OPT_{\text{MaxLBP-U}}(V)|$ with high probability, under uniform or linear power assignment. Under half-duplex communication, an additional logarithmic factor in the running time ensures the same high probability approximation factor.*

9.8 Chapter Summary

In this chapter, we present the first set of fast distributed algorithms in the SINR model with a constant factor approximation guarantee for MaxLSP-U, and the MaxLBP problem with uniform-power data transmission but non-uniform ranges. We extensively study the problem by accounting for the cases of half/full duplex and non-adaptive/adaptive power availability for scheduling. The non-local nature of this model and the asymmetry between senders and receivers makes this model very challenging to study. Our algorithm is randomized and crucially relies on physical carrier sensing for the distributed communication steps, without any additional assumptions. Our main technique of distributed computation of a ruling is likely to be useful in the design of other distributed algorithms in the SINR model.

9.9 Appendix to Section 9.3

9.9.1 Proof of Lemma 9.1

The statement of Lemma 9.1 is equivalent to that $\forall l \in S, A(S \setminus \{l\}, l) \le 1$. Let l be an arbitrary link in S, and w.l.o.g., we assume $l \in S_i$, and thus $l \in L_i$. In each phase $j < i$, because $\overline{J_j^a} \cup \overline{J_j^b} \cup J_j^r \cup J_j^z \supseteq J_j$, all the links in $\cup_{j<i} L_j$ have been removed from J at the end of phase j. Due to the 2nd step, $A(\cup_{j<i} S_j, \overleftarrow{l}) \le \psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^{\alpha}$. First, we show that $A(\cup_{j<i} S_j, l) \le \psi$.

For any link $l' \in \cup_{j<i} S_j$,

$$A(l', \overleftarrow{l}) = \frac{\beta}{1 - \frac{d^\alpha(\overleftarrow{l})}{P/(\beta N)}} \frac{d^\alpha(\overleftarrow{l})}{d^\alpha(l', \overleftarrow{l})} \leq A(\cup_{j<i} S_j, \overleftarrow{l}) < 1.$$

Hence, $d(x(l'), x(l)) = d(l', \overleftarrow{l}) \geq \big(\frac{\beta d^\alpha(l)}{1 - \frac{d^\alpha(l)}{P/(\beta N)}}\big)^{1/\alpha} \geq (\frac{\beta(1+\phi)}{\phi})^{1/\alpha} d(l)$, implying $\frac{d(l',l)}{d(l',\overleftarrow{l})} = \frac{d(x(l'),r(l))}{d(x(l'),x(l))} \geq \frac{d(x(l'),x(l)) - d(l)}{d(x(l'),x(l))} \geq 1 - (\frac{\phi}{\beta(1+\phi)})^{1/\alpha}$. By referring to the definition of affectance — $A(\cup_{j<i} S_j, l) = \sum_{l' \in \cup_{j<i} S_j} \frac{\beta}{1 - \frac{d^\alpha(l)}{P/(\beta N)}} \frac{d^\alpha(l)}{d^\alpha(l',l)}$ and $A(\cup_{j<i} S_j, \overleftarrow{l}) = \sum_{l' \in \cup_{j<i} S_j} \frac{\beta}{1 - \frac{d^\alpha(\overleftarrow{l})}{P/(\beta N)}} \frac{d^\alpha(\overleftarrow{l})}{d^\alpha(l', \overleftarrow{l})}$ — we obtain

$$A(\cup_{j<i} S_j, l) \leq \big(\frac{1}{1 - (\frac{\phi}{\beta(1+\phi)})^{1/\alpha}}\big)^\alpha A(\cup_{j<i} S_j, \overleftarrow{l}) = \psi.$$

Next, it suffices to show that

$$A(\cup_{j\geq i} S_j \setminus \{l\}, l) \leq 1 - \psi.$$

At phase i, $\omega_1 = \gamma_1 d_i$; the 1st step in the phase ensures that when link l is added to S_i, any link $l' \in \cup_{j\geq i} L_j$ with $d(x(l'), x(l)) < \omega_1$ will not get in S_i. Therefore, for the set $\cup_{j\geq i} S_j \setminus \{l\}$, we have: (1) all the nodes in $X(\cup_{j\geq i} S_j \setminus \{l\})$ have a mutual distance of at least ω_1; (2) the distance from any node in $X(\cup_{j\geq i} S_j \setminus \{l\})$ to $r(l)$ is at least $\omega_1 - d(l)$; and (3) $\omega_1 - d(l) > \omega_1/2 > 0$. According to Proposition 9.13, by using $\gamma_1 = \big(\frac{36\beta}{1-\psi} \frac{\alpha-1}{(\alpha-2)} \frac{1+\phi}{\phi}\big)^{1/\alpha} + 2$, $SP\big(X(\cup_{j\geq i} S_j \setminus \{l\}), r(l)\big) \leq \frac{36(\alpha-1)}{\alpha-2} \frac{P}{(\omega_1 - d(l))^\alpha} + N < \frac{(1-\psi)P}{\beta d^\alpha(l)}(1 - \frac{d^\alpha(l)}{P/(\beta N)}) + N$. It is easy to verify $A(\cup_{j\geq i} S_j \setminus \{l\}, l) < 1 - \psi$.

Proposition 9.13. *$\forall V' \in V$ and $\forall v \notin V'$, if (1) all the nodes in V' are at least ρ_1 away from each other, (2) the distance between v and any node in V' is at least ρ_2, (3) $\rho_2 > \rho_1/2 > 0$, then $SP(V', v) < \frac{36(\alpha-1)}{\alpha-2} \frac{\rho_2^2}{\rho_1^2} \frac{P}{\rho_2^\alpha} + N$.*

Proof. We bound the sensed power strength by partitioning the plane into concentric rings all centered at v, each of width ρ_2, via a similar technique to that in [25, 103]. Let $Ring(i)$ denote the ith ring (where $i = 1, 2, \ldots$), which contains every node v' that satisfies $i\rho_2 \leq d(v', v) < (i+1)\rho_2$; let $V'(i)$ denote the subset of nodes in V' that fall in $Ring(i)$. We notice the following facts: (1) For any two nodes $v, v' \in V'(i)$, two disk centered at v, v' respectively with a radius of $\rho_1/2$ are non-overlapping. (2) For any node $v \in V'(i)$, such a disk is fully contained in an extended ring $Ring'(i)$ of $Ring(i)$, with an extra width of $\rho_1/2$ at each side of $Ring(i)$. The area (denoted by D) of each of such disks is $D = \pi(\rho_1/2)^2$. The area (denoted by $D(i)$) of $Ring'(i)$ is

$$D(i) = \pi[((i+1)\rho_2 + \rho_1/2)^2 - (i\rho_2 - \rho_1/2)^2] \leq 3\pi(2i+1)\rho_2^2.$$

Using $|V'(i)| \leq D(i)/D \leq 12(2i+1)\rho_2^2/\rho_1^2$, we obtain

$$\begin{aligned} SI(V', w) &\leq \sum_{i=1}^{\infty} |V'(i)| \frac{P}{(i\rho_2)^\alpha} + N \\ &\leq \sum_{i=1}^{\infty} \frac{12(2i+1)}{i^\alpha} \frac{\rho_2^2}{\rho_1^2} \frac{P}{\rho_2^\alpha} + N \\ &\leq \frac{36(\alpha-1)}{\alpha-2} \frac{\rho_2^2}{\rho_1^2} \frac{P}{\rho_2^\alpha} + N. \qquad \square \end{aligned}$$

9.9.2 Proof of Theorem 9.2

For a node v, we define $B(v, d)$ as the ball centered at v with a radius of d. With a parameter $\gamma > 1$, we then define a link set $B_\gamma^{\geq}(l)$, such that for a link $l \in L_i$, $B_\gamma^{\geq}(l)$ contains all and only the links in $\{L_j : j \geq i\}$, with their senders in the ball $B(x(l), \gamma d_i)$; in other words, $B_\gamma^{\geq}(l)$ contains the links with similar or longer lengths, whose senders are (γd_i)-covered by $x(l)$. For a set $L' \subseteq L$, $B_\gamma^{\geq}(L')$ is defined as $\cup_{l \in L'} B_\gamma^{\geq}(l)$.

Lemma 9.14 (Spatial Constraint)**.** *$\forall \gamma > 1$, $\forall L' \subseteq L$, $\forall l \in L'$, $|OPT(B_\gamma^{\geq}(l) \cap L')| \leq C_1(\gamma)$, where $C_1(\gamma) = \frac{(2\gamma+1)^\alpha}{\beta}$.*

Proof. Let k be an arbitrary link in $OPT(B_\gamma^{\geq}(l) \cap L')$. For any link $k' \in OPT(B_\gamma^{\geq}(l) \cap L')$,

$d(k', k) \leq d(x(k'), x(l)) + d(x(l), x(k)) + d(k) \leq (2\gamma + 1)d(k)$. Therefore,

$$\begin{aligned}
1 &\geq A(OPT(B_{\gamma}^{\geq}(l) \cap L') \setminus \{k\}, k) \\
&= \beta \frac{\sum_{k' \in OPT(B_{\gamma}^{\geq}(l) \cap L') \setminus \{k\}} \frac{d^{\alpha}(k)}{d^{\alpha}(k',k)}}{1 - \frac{d^{\alpha}(k)}{P/(\beta N)}} \\
&\geq \beta \frac{|OPT(B_{\gamma}^{\geq}(l) \cap L')| \frac{d^{\alpha}(k)}{(2\gamma+1)^{\alpha} d^{\alpha}(k)}}{1 - \frac{d^{\alpha}(k)}{P/(\beta N)}} \\
&\geq \beta(2\gamma + 1)^{-\alpha} |OPT(B_{\gamma}^{\geq}(l) \cap L')|.
\end{aligned}$$

It follows that $|OPT(B_{\gamma}^{\geq}(l) \cap L')| \leq (2\gamma + 1)^{\alpha}/\beta = C_1(\gamma)$. □

Lemma 9.15 (Affectance Constraint)**.** *$\forall \psi' > 0$ and $\forall L', L'' \subseteq L$, if $L' \cap L'' = \emptyset$ and $A(L', \overleftarrow{l}) > \psi'$ for any link $l \in L''$, then $|OPT(L'')| \leq C_2(\psi')|L'|$, where $C_2(\psi') = (\frac{2(\beta b)^{1/\alpha}}{(\beta b)^{1/\alpha} - 1})^{\alpha}/\psi' + 1$.*

Proof. If $|OPT(L'')| > 0$, we can express it as $|OPT(L'')| = b|L'| + g$, such that b and g are non-negative integers and $1 \leq g \leq |L'|$. We create $|L'|$ bins, each of which has a capacity of b links; we pack the links in $OPT(L'')$ to the bins via a first-fit sweep through the links in L':

(1) We order the links in L' arbitrarily; let l_j denote the jth link in L', and let Bin_j denote the jth bin. Let set $L^* = OPT(L'')$ initially; then, the sweep proceeds in $|L'|$ rounds.

(2) In the ith round (where $i = 1, 2, \ldots, |L'|$), we pick b links in L^* whose senders are the nearest b nodes to the sender $x(l_i)$ of the ith link in L', and we remove those b links from L^* and put them into Bin_i.

The completion of the above packing means that in $OPT(L'')$, we have $b|L'|$ links "near", and $g \in [1, |L'|]$ links "far" from the senders of links in L'. If $b \leq (\frac{2(\beta b)^{1/\alpha}}{(\beta b)^{1/\alpha} - 1})^{\alpha}/\beta$, we are done. Therefore, for the rest of the proof we assume that $b > (\frac{2(\beta b)^{1/\alpha}}{(\beta b)^{1/\alpha} - 1})^{\alpha}/\beta$, and we show that $b \leq C_2(\psi') - 1 = (\frac{2(\beta b)^{1/\alpha}}{(\beta b)^{1/\alpha} - 1})^{\alpha}/\psi'$ in this case.

Let l denote a "far" link in $OPT(L'')$ that is out of any bins. We have for any link $k \in Bin_i, d(x(k), x(l_i)) < d(x(l), x(l_i))$, implying that $d(x(l), x(k)) < 2d(x(l), x(l_i))$ due to triangle inequality. Since $Bin_i \cup \{l\} \subseteq OPT(L'')$, we have

$$
\begin{aligned}
1 \geq A(Bin_i, l) &= \beta \frac{\sum_{k \in Bin_i} \frac{d^\alpha(l)}{d^\alpha(k,l)}}{1 - \frac{d^\alpha(l)}{P/(\beta N)}} \\
&\geq \beta \frac{b\big(\frac{d(l)}{2d(x(l),x(l_i))+d(l)}\big)^\alpha}{1 - \frac{d^\alpha(l)}{P/(\beta N)}} \\
&\geq \beta b \big(\frac{1}{2d(x(l),x(l_i))/d(l)+1}\big)^\alpha.
\end{aligned}
$$

That leads to that $d(l) \leq \frac{2}{(\beta b)^{1/\alpha}-1} d(x(l), x(l_i))$. Then for any link $k \in Bin_i$,

$$
\begin{aligned}
& d(k,l) = d(x(k), r(l)) \\
\leq & d(x(k), x(l_i)) + d(x(l_i), x(l)) + d(x(l), r(l)) \\
\leq & \frac{2(\beta b)^{1/\alpha}}{(\beta b)^{1/\alpha} - 1} d(x(l_i), x(l)).
\end{aligned} \tag{9.1}
$$

Since $\cup_i Bin_i \cup \{l\} \subseteq OPT(L'')$, we have

$$
\begin{aligned}
1 \geq A(\cup_{i=1}^{|L'|} Bin_i, l) &= \beta \sum_{i=1}^{|L'|} \frac{\sum_{k \in Bin_i} \frac{d^\alpha(l)}{d^\alpha(k,l)}}{1 - \frac{d^\alpha(l)}{P/(\beta N)}} \\
&\geq \beta \sum_{i=1}^{|L'|} \frac{\frac{|Bin_i| d^\alpha(l)}{(\frac{2(\beta b)^{1/\alpha}}{(\beta b)^{1/\alpha}-1})^\alpha d^\alpha(x(l_i),x(l))}}{1 - \frac{d^\alpha(l)}{P/(\beta N)}} \quad \text{(by Ineq. (9.1))} \\
&\geq \frac{b}{(\frac{2(\beta b)^{1/\alpha}}{(\beta b)^{1/\alpha}-1})^\alpha} \frac{\beta \sum_{l' \in L'} \frac{d^\alpha(l)}{d^\alpha(x(l'),x(l))}}{1 - \frac{d^\alpha(l)}{P/(\beta N)}} \\
&= b\Big(\frac{(\beta b)^{1/\alpha} - 1}{2(\beta b)^{1/\alpha}}\Big)^\alpha A(L', \overleftarrow{l}) \\
&\geq b\Big(\frac{(\beta b)^{1/\alpha} - 1}{2(\beta b)^{1/\alpha}}\Big)^\alpha \psi'.
\end{aligned}
$$

The last inequality above holds because

$$
A(L', \overleftarrow{l}) = \beta \frac{\sum_{l' \in L'} \frac{d^\alpha(l)}{d^\alpha(x(l'),x(l))}}{1 - \frac{d^\alpha(l)}{P/(\beta N)}} \geq \psi'.
$$

Therefore, $b \leq (\frac{2(\beta b)^{1/\alpha}}{(\beta b)^{1/\alpha}-1})^{\alpha}/\psi'$ and $|OPT(L'')| \leq (b+1)|L'| \leq C_2(\psi')|L'|$. $\square$

We define $J^a = \cup_i J_i^a, \overline{J^a} = \cup_i \overline{J_i^a}, J^b = \cup_i J_i^b, \overline{J^b} = \cup_i \overline{J_i^b}, J^r = \cup_i J_i^r, J^z = \cup_i J_i^z$. $\overline{J^a} \cup \overline{J^b}$ contains all the links removed in the 1st step in Algorithm 2 due to the affectance constraints. At each phase i, $X(J_i^r)$ is an (ω_1, ω_2)-ruling of $X(J_i^a)$, and all the nodes in $X(J_i^z)$ are ω_2-covered by $X(J_i^r)$; we choose all the links in J_i^r to add to S, discard all the links in $\overline{J_i^a} \cup \overline{J_i^b}$ (for failing affectance check), and also discard all the links in J_i^z (because of their proximity to the chosen links).

We have

$$L = \cup_i(\overline{J_i^a} \cup \overline{J_i^b} \cup J_i^r \cup J_i^z) \subseteq \cup_i(\overline{J_i^a} \cup \overline{J_i^b}) \cup \cup_i B_{\gamma_2}^{\geq}(J_i^r) = \overline{J^a} \cup \overline{J^b} \cup B_{\gamma_2}^{\geq}(S).$$

Due to Lemma 9.15, $|OPT(\overline{J^a} \cup \overline{J^b}| \leq C_2(\psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^{\alpha})|S|$; due to Lemma 9.14, $|OPT(B_{\gamma_2}^{\geq}(S))| \leq \sum_{l \in S} |OPT(B_{\gamma_2}^{\geq}(l))| \leq C_1(\gamma_2)|S|$. Therefore,

$$\begin{aligned} &|OPT(L)| \leq |OPT(\overline{J^a} \cup \overline{J^b})| + |OPT(B_{\gamma_2}^{\geq}(S))| \\ \leq &\big(C_1(\gamma_2) + C_2(\psi(1-(\frac{\phi}{\beta(1+\phi)})^{1/\alpha})^{\alpha})\big)|S|. \end{aligned}$$

9.10 Appendix to Section 9.4

Recall that we call an iteration of the outer loop (Line 1) a phase of the algorithm, and an iteration of the inner loop (Line 2) a round; recall that $B(v,d)$ denotes the ball centered at v with a radius of d. Let $A_t^{W_1}(v,d)$ denote the set of *active* nodes in set W_1 that fall in the ball $B(v,d)$ at time point t; we will explicitly point t out whenever we use $A_t^{W_1}(v,d)$.

9.10.1 Proof of Lemma 9.4

The following definitions are only involved in Lemmas 9.16 and 9.17. Let η be a constant $> \left(96\frac{\alpha-1}{\alpha-2}\right)^{-1/\alpha}$. In one round (which corresponds to one iteration of the inner loop) of Algorithm 3, let $U \in W_1$ be the set of nodes with $U() = 1$ at line 5 in the coordination step. We say an active node $v \in W_1$ is "lucky" in a round, with t_0 being the time that the round starts, if and only if

(1) $v \in U$;

(2) $U \cap A^{W_1}_{t_0}(v, \eta\omega_1) = \{v\}$, *i.e.*, v has no nearby active nodes in U;

(3) $SP(U \setminus A^{W_1}_{t_0}(v, \eta\omega_1), v) < Thres(\omega_1)$, *i.e.*, total power received from faraway active nodes is small.

In a round if v gets lucky, $U(v)$ will remain 1 till the end of that round, and thus will elect to be included in $\hat{R}$ and will cause all nodes in $A^{W_1}_{t_0}(v, \omega_1)$ to get into $\hat{Z}$.

Lemma 9.16. *In a round i_{in} of phase i_{out}, with t_1 being the time that the round starts, suppose that for each active node $u \in W_1, |A^{W_1}_{t_1}(u, \omega_1)| \leq 2^{\log b_{max} - i_{out} + 1}$ at the beginning of the round, then the probability for an arbitrary active node in W_1 to be lucky in the round is at least $2^{-(\log b_{max} - i_{out} + 3 + (2\eta+1)^2)}$.*

Proof. For a round at phase i_{out}, we prove the statement in the following four steps.

(1) First, for any active node $v \in W_1$, the probability for v to be in U is $Prob\big(U(v) = 1\big) = 2^{-(\log b_{max} - i_{out} + 2)}$.

(2) Second, for any active node $v \in W_1$, due to the packing property we upper-bound the size of $A^{W_1}_{t_1}(v, \eta\omega_1)$ as

$$\big|A^{W_1}_{t_1}(v, \eta\omega_1)\big| \leq \frac{\pi(\eta\omega_1 + \omega_1/2)^2}{\pi(\omega_1/2)^2} \max_{v'} \big|A^{W_1}_{t_1}(v', \omega_1)\big| = (2\eta+1)^2 2^{\log b_{max} - i_{out} + 1}.$$

Then, because $\log b_{max} + 1 \geq i_{out}$, $2^{\log b_{max} - i_{out} + 2} \geq 2$. the probability for all nodes other than v in $A^{W_1}_{t_1}(v, \eta\omega_1)$ to not appear in U (*i.e.*. to remain silent) is

$$\begin{aligned} & Prob\big(A^{W_1}_{t_1}(v, \eta\omega_1) \cap U \setminus \{v\} = \emptyset\big) \\ \geq & \prod_{u \in A^{W_1}_{t_1}(v, \eta\omega_1)} \big(1 - Prob(U(u) = 1)\big) \geq \big(1 - 2^{-(\log b_{max} - i_{out} + 2)}\big)^{|A^{W_1}_{t_1}(v, \eta\omega_1)|} \\ \geq & \big(1 - 2^{-(\log b_{max} - i_{out} + 2)}\big)^{(2\eta+1)^2 2^{\log b_{max} - i_{out} + 1}} \geq (1/4)^{(2\eta+1)^2/2} = 2^{-(2\eta+1)^2}. \end{aligned}$$

(3) Third, for any active node $v \in W_1$, we lower-bound the probability that v's received power $SP(U \setminus A^{W_1}_{t_1}(v, \eta\omega_1), v)$ from outside of the ball $B(v, \eta\omega_1)$ is "low" — *i.e.*, below $Thres(\omega_1)$ — by (i) partitioning the plane into concentric rings via a similar technique to that in [25, 103], and (ii) referring to an $(\eta\omega_1, \eta\omega_1)$-ruling and determining the expected number of nodes in each a ring that appear in U, so that we can bound the power received

from all the nodes in the rings outside of $B(v, \eta\omega_1)$.

We partition the plane into rings all centered at v, each of width $\eta\omega_1$. Let $Ring(h)$ denote the hth ring, which contains every node v' that satisfies $h\eta\omega_1 \leq d(v', r(l)) < (h+1)\eta\omega_1$, for each $h = 1, 2, \ldots$; let $Ring^a(h)$ denote the set of active nodes in $Ring(h)$. When $h = 0$, $Ring(0)$ corresponds to the ball $B(v, \eta\omega_1)$. Let $R(h)$ denote an $(\eta\omega_1, \eta\omega_1)$-ruling of $Ring^a(h)$. Then by noticing that (i) $Ring^a(h) \subseteq \cup_{v' \in R(h)} A_{t_1}^{W_1}(v', \eta\omega_1)$, and (ii) for any two nodes $v', u' \in R(h), d(v', u') > \eta\omega_1$, we have

$$\begin{aligned} & \mathbb{E}\left\{\left|U \cap Ring^a(h)\right|\right\} \\ = & \sum_{v' \in Ring^a(h)} \mathbb{E}\{U(v')\} \leq \sum_{v' \in R(h)} \sum_{u' \in A_{t_1}^{W_1}(v', \eta\omega_1)} \mathbb{E}\{U(u')\} \\ = & \sum_{v' \in R(h)} \left|A_{t_1}^{W_1}(v', \eta\omega_1)\right| \frac{b_{max}}{2^{i_{out}-2}} \leq \left|R(h)\right|/2. \end{aligned} \tag{9.2}$$

To bound the cardinality of $R(h)$, we use the following facts: (i) for any two nodes $v', u' \in R(h)$, two disk centered at v', u' respectively with a radius of $\eta\omega_1/2$ are non-overlapping; and (ii) For any node $v' \in Ring^a(h)$, such a disk is fully contained in an extended ring $Ring'(h)$ of $Ring(h)$, with an extra width of $\eta\omega_1/2$ at each side of $Ring(h)$. Then, by referring to the ratio between the areas of $Ring'(h)$ and a disk, we have $\left|R(h)\right| \leq 8(2h+1)$; Inequality (9.2) yields $\mathbb{E}\left\{\left|U \cap Ring^a(h)\right|\right\} \leq 2^4(2h+1)$.

Therefore, v's received power from outside of the ball $B(v, \eta\omega_1)$ is

$$\begin{aligned} & \mathbb{E}\{SP(U \setminus A_{t_1}^{W_1}(v, \eta\omega_1), v)\} \\ = & \mathbb{E}\Big\{\sum_{h=1}^{\infty} \sum_{v' \in U(h)} \frac{P'}{d^\alpha(v', v)} + N\Big\} \\ \leq & \sum_{h=1}^{\infty} \mathbb{E}\left\{\left|U \cap Ring^a(h)\right|\right\} \frac{P'}{(h\eta\omega_1)^\alpha} + N \\ \leq & \sum_{h=1}^{\infty} \frac{2^4(2h+1)}{h^\alpha} \frac{P'}{(\eta\omega_1)^\alpha} + N \\ \leq & \frac{48}{\eta^\alpha} \frac{\alpha-1}{\alpha-2} \frac{P'}{\omega_1^\alpha} + N \leq Thres(\omega_1)/2. \end{aligned}$$

According to Markov's Inequality,

$$Prob\big(SP(U \setminus A_{t_1}^{W_1}(v, \omega_2), v) \geq Thres(\omega_1)\big) \leq 1/2,$$

implying

$$Prob\big(SP(U \setminus A_{t_1}^{W_1}(v, \omega_2), v) < Thres(\omega_1)\big) \geq 1/2.$$

(4) Finally, combining the above three, the probability that v is lucky is at least

$$2^{-(\log b_{max} - i_{out} + 3 + (2\eta+1)^2)}. \qquad \square$$

Lemma 9.17. *In a round i_{in} of phase i_{out}, with t_1 being the time that the round starts, suppose that for each active node $u \in W_1$, $\left|A_{t_1}^{W_1}(u, \omega_1)\right| \leq 2^{\log b_{max} - i_{out} + 1}$; then, for an arbitrary active node $v \in W_1$ with $\left|A_{t_1}^{W_1}(v, \omega_1)\right| \geq 2^{\log b_{max} - i_{out}}$, the probability that v becomes inactive by the end of the round is at least a constant C_6, where $0 < C_6 = 2^{-(3+(2\eta+1)^2)} < 1$.*

Proof. In a round of phase i_{out}, a sufficient condition for v to be inactive by the end of the round is that either v or any node in $A_{t_1}^{W_1}(v, \omega_1)$ enters $\hat{R}$, such that v either enters $\hat{R}$ or $\hat{Z}$ and exits the algorithm. Further, that either v or any node in $A_{t_1}^{W_1}(v, \omega_1)$ gets lucky satisfies this condition. Therefore, the probability for v to become inactive in the round is at least

$$\sum_{v' \in A_{t_1}^{W_1}(v,\omega_1)} Prob(v' \text{ is lucky}) \geq \left|A_{t_1}^{W_1}(v, \omega_1)\right| 2^{-\left(\log b_{max} - i_{out} + 3 + (2\eta+1)^2\right)} \geq 2^{-\left(3+(2\eta+1)^2\right)}. \qquad \square$$

Lemma 9.18. *Let $E_{i_{out}}$ denote the event that at the end of the phase i_{out}, $\left|A_{t_1}^{W_1}(u, \omega_1)\right| \leq b_{max}/2^{i_{out}}$ for every active node $u \in W_1$, where t_1 is the time that the last round in i_{out} ends. $\forall i_{out}$, $Prob(E_{i_{out}}) \geq 1 - i_{out}/n^{C_7}$, for some positive constant C_7.*

proof by induction. At the end of the phase $i_{out} = 0$, this is trivial. Suppose that for $i_{out} = i \geq 1$ the statement is true; we show that it still holds for $i_{out} = i + 1$. At the beginning of phase $i + 1$, if we already have event E_{i+1}, we are done; otherwise, let v denote an active node such that $\left|A_{t_2}^{W_1}(v, \omega_1)\right| > \frac{b_{max}}{2^{i+1}}$, where t_2 is the time that the first round in $i + 1$ starts. We show $Prob(E_{i+1}) \geq 1 - \frac{i+1}{n^{C_7}}$ as below:

(1) We choose a constant $C_4 \geq \frac{C_7+1}{\log \frac{1}{1-C_6}}$ for the inner loop at Line 2 of Algorithm 3.

(2) Under the induction assumption for phase i, we have: for any round i_{in} during phase $i+1$,

with t_3 being the time that this round starts, if $\left|A_{t_3}^{W_1}(v,\omega_1)\right| \leq \frac{b_{max}}{2^{i+1}}$, we are done; otherwise, as long as $\frac{b_{max}}{2^i} \geq \left|A_{t_3}^{W_1}(v,\omega_1)\right| \geq \frac{b_{max}}{2^{i+1}}$, the probability for v to turn inactive during the round is at least C_6 according to Lemma 9.17. Then, the probability for v to become inactive by the end of phase $i+1$ (consisting of $C_4 \log n$ rounds) is at least $1-(1-C_6)^{C_4 \log n} \geq 1-1/n^{C_7+1}$.

(3) By considering both the conditional probability and the fact that there are at most n such nodes as v,

$$Prob(E_{i+1}) \geq Prob(E_{i+1} \mid E_i)Prob(E_i) \geq (1-\frac{n}{n^{C_7+1}})(1-\frac{i}{n^{C_7}}) \geq 1-\frac{i+1}{n^{C_7}}. \qquad \square$$

At the end of phase $i_{out} = \log b_{max}+1$, with t_4 being the time that the last round in phase i_{out} ends, we have that with probability of at least $1-\frac{\log b_{max}+1}{n^{C_7}}$, for every active node $u \in W_1$, $|A_{t_4}^{W_1}(u,\omega_1)| \leq 1/2 < 1$. That means that all the nodes in W_1 have joined either $\hat{R}$ or $\hat{Z}$, with probability of at least $1-\frac{1}{n^{C_7-1}}$, concluding the proof of Lemma 9.4.

9.10.2 Proof of Lemma 9.7

Proof by contradiction. Suppose there exists a node $v \in \hat{Z}$ that is not ω_2-covered by $\hat{R}$. Therefore, all the nodes in $\hat{R}$ are outside of the ball $B(v,\omega_2)$. We calculate $SP(\hat{R},v)$ and derive the conflict. We notice the following three facts: (1) all the nodes in $\hat{R}$ have a mutual distance of at least ω_1; (2) the distance from any sender node in $\hat{R}$ to v is $> \omega_2$; and (3) $\omega_2 > \omega_1/2 > 0$. Due to Proposition 9.13,

$$SP(\hat{R},v) \leq 36\frac{\alpha-1}{\alpha-2}\frac{P}{\omega_1^2(\omega_2)^{\alpha-2}} + N \leq 36\frac{\alpha-1}{\alpha-2}\frac{\omega_1^{\alpha-2}}{\omega_2^{\alpha-2}}Thres(\omega_1) \leq Thres(\omega_1).$$

According to the condition for v to enter $\hat{Z}$, we must have $SP(\hat{R},v) > Thres(\omega_1)$, where lies the contradiction.

9.10.3 Proof of Lemma 9.8

Suppose there is "bad" node $v \in \hat{R}$ such that there exists a node $v' \in \hat{R}$ and $d(v,v') < \omega_1$. We call such a v' a "bad" partner of v. The only possible situation for v to have a "bad" partner v' is that v and v' enter $\hat{R}$ in the same round; otherwise, one of them should have been "pushed" into $\hat{Z}$ during the decision step of a round when the other enters $\hat{R}$.

Let u denote an arbitrary node in $\hat{R}$. The necessary and sufficient condition for u to be bad (or to have a bad partner) is that, in the coordination step of the round when u enters $\hat{R}$, there exists at least one active node $u' \in W_1$ such that, (1) $d(u,u') < \omega_1$, (2) u and u' are in U, and (3) u and u' made the same random binary decisions all through the $C_5 \log m$ slots of the coordination step.

W.l.o.g., we assume u enters $\hat{R}$ at round i_{in} in phase i_{out}. Let t_1 denote the time that round i_{in} in phase i_{out} begins. The probability that u is bad equals the probability for at least one other active node $u' \in A_{t_1}^{W_1}(u,\omega_1)$ (*i.e.*, $d(v,v') < \omega_1$) to enter $\hat{R}$ at round i_{in} in phase i_{out}.

$$\begin{aligned}
& Prob(u \text{ is bad}) \\
\leq & \sum_{u' \in A_{t_1}^{W_1}(u,\omega_1)} Prob(U(u')=1) \frac{1}{2^{C_5 \log n}} \\
\leq & \frac{1}{n^{C_5}} \frac{\left|A_{t_1}^{W_1}(u,\omega_1)\right|}{2^{\log b_{max} - i_{out} + 2}} \\
\leq & \frac{1}{n^{C_5}} \frac{\max_{\text{active } w \in W_1} \left|A_{t_1}^{W_1}(w,\omega_1)\right|}{2^{\log b_{max} - i_{out} + 2}}.
\end{aligned}$$

Further, due to Lemma 9.18,

$$Prob\Big(\max_{\text{active } w \in W_1} \left|A_{t_1}^{W_1}(w,\omega_1)\right| \leq 2^{\log b_{max} - i_{out} + 1}\Big) \geq 1 - \frac{i_{out} - 1}{n^{C_7}} \geq 1 - \frac{1}{n^{C_7 - 1}}.$$

Therefore,

$$Prob(u \text{ is good}) \geq (1 - \frac{1}{n^{C_5}})(1 - \frac{1}{n^{C_7 - 1}}) \geq 1 - \frac{2}{n^{\min\{C_5, C_7 - 1\}}}.$$

Finally, since $\hat{R}$ contains at most n nodes, the probability that there are no bad nodes in $\hat{R}$ is at least $1 - \frac{1}{n^{\min\{C_5, C_7 - 1\} - 1}}$.

Chapter 10

Fast Distributed Algorithms for Constructing Non-uniform-power MIS under the Physical Interference Model

10.1 Preliminaries and Definitions

Apart from the definitions and symbols used for MaxLSP-U in the previous chapter, we introduce *local density* denoted by ρ as the maximum number of links in the same link class L_i such that their sender nodes fall in the same disk of radius d_i.

10.2 Problem Definition: The Maximum Link Scheduling Problem (MaxLSP)

Given a set of communication requests (links) L, and a power level $P(l)$ for each link $l \in L$, the goal is to find a largest independent subset (that can be scheduled simultaneously in the SINR model). We assume the power level assignment is length-monotone and sub-linear, which means that for any links l, l', $d(l') \geq d(l)$ implies $P(l') \geq P(l)$ and $\frac{P(l')}{d^\alpha(l')} \leq \frac{P(l)}{d^\alpha(l)}$. In this chapter, we use $OPT^P(L)$ to denote an optimum solution for MaxLSP, *i.e.*, the cardinality of the largest such independent set for the given length-monotone and sub-linear power assignment; we drop the super-script P to denote the underlying power assignment, whenever it is clear from the context. As discussed earlier, computing $OPT(L)$ is NP-hard, and we focus on approximation algorithms. We say an algorithm gives a C-approximation factor if it constructs an independent link set $L' \subseteq L$ with $|L'| \geq |OPT(L)|/C$.

10.3 Distributed Algorithm for MaxLSP

In this section, we present the distributed algorithm for MaxLSP under a length-monotone sublinear power assignment. In order to simplify the discussion, we first explain the centralized algorithm of [53] and the main ideas of its analysis. We then discuss the distributed implementation.

10.3.1 The Centralized Algorithm

The centralized algorithm [53] for MaxLSP forms the basis for the distributed algorithm. The algorithm processes links in non-decreasing order of length. Recall the notion of affectance from Section 10.1. Let $L = \{l_1, \ldots, l_n\}$ be the initial set of links in non-decreasing order of length. Let S be the set of links already chosen (which is empty initially). For link l: if $A(S, l) + A(l, S) < 1/2$, add l to S. After considering all links, the set $S' = \{l \in S : A(S, l) \leq 1\}$ is selected as the solution after verifying the affectance.

By construction, the solution S' is feasible. The main idea in proving the approximation factor of the solution is the following "Red-Blue Lemma" from [53]. For a set Y of links, define $Y^+(l) = \{l' \in Y : d(l') \geq d(l)\}$ and $Y^-(l) = \{l' \in Y : d(l') \leq d(l)\}$.

Lemma 10.1. *(Red-Blue Lemma [53]) Let $RED, BLUE$ be disjoint link sets. If $|BLUE| > 4|RED|$ and $BLUE$ is a 3^α-signal set, then $\exists$ link $l \in BLUE$ s.t. $A(RED^-(l), l) + A(l, RED^-(l)) \leq 3^\alpha(A(BLUE, l) + A(l, BLUE))$ under any length-monotone sublinear power assignment.*

Designing an algorithm that filters links based on global interference constraint in a distributed manned under the physical interference that is non-local, non-linear and cumulative is challenging. Any sub-linear time distributed algorithm requires selecting and removing a set of links in parallel: while ensuring that they form an independent set together with all those previously selected, we need to avoid over-trimming the links. This is achieved in our algorithm critically via the use of carrier sensing. Another complication comes from the sender-receiver separation that requires controlling the interference on the receivers from the senders' perspective.

10.3.2 Additional Definitions: Cover, Ruling and Restricted Ruling

We adopt the definitions of cover and ruling from Section 9.3.2 in Chapter 9. The concept of ruling has a vital role in our algorithm: it is used for choosing a set of spatially separated links and removing the nearby links of the chosen links. Based on that, we further define a restricted ruling as follows. Let X be an arbitrary subset of senders and R be an arbitrary subset of receivers of links in the same link class. We define a *R-restricted* (ω_1, ω_2)*-ruling* of X as $R_{\omega_1,\omega_2}(X')$, where $X' = \{v \in X : v\text{'s corresponding receiver node is in } R\}$.

10.3.3 The Distributed Algorithm

Algorithm 7 describes the distributed scheduling algorithm for MaxLSP with any length-monotone sublinear power assignment. We use the following constants in the algorithm $\gamma_1 = \left(\frac{36\beta}{1-\psi}\frac{\alpha-1}{(\alpha-2)}\frac{1+\phi}{\phi}\right)^{1/\alpha} + 2$ and γ_2 as an arbitrary constant $> \gamma_1$, where α, β, ϕ are constants described in Inequality (8.1) and ψ is a constant that can take any value from $(0, 1)$. The algorithm consists of two steps: (Step 1) selecting links according to affectance and ruling constraints (Lines 2-16), and (Step 2) verifying affectance constraints (Lines 17-24). In each step, it sweeps through the link classes in $g(L)$ phases. The lemmas in this section and Theorem 10.6 are true under any length-monotone sublinear power assignment.

In the ith phase of Step 1, where $i \in [1, g(L)]$, it selects a subset of links S_i from L_i. At Lines 3-13, both the sender and the receiver of each link in L_i estimates and checks the affectance to and from selected links in $\cup_{j<i} S_j$, and obtain the sets X_i and R_i of nodes that survive this process. Then, at Line 15 we compute a restricted ruling R_i^r using Algorithm 8, such that R_i^r is an (ω_1, ω_2)-ruling of $\{x(k) : x(k) \in X_i \text{ and } r(k) \in R_i, \forall k \in L\}$, where $\omega_1 = \gamma_1 d_i$ and $\omega_2 = \gamma_2 d_i$. Then S_i is the set of links whose sender nodes are in R_i^r. For each link $l \in S_i$, due to the confirmation mechanism (Lines 10-14 in Algorithm 8) in the construction of R_i^r, the receiver $r(l)$ is also aware of that $x(l) \in R_i^r$. Therefore, both the

sender and the receiver of each $l \in S_i$ knows the status of l, so that Lines 4 and 8 are valid distributed operations. Lemma 10.3 is a direct result of the definition of ruling and the choice of ω_1, ω_2.

Proposition 10.2. *For any subset L' of L_i, $\forall i$, if all the sender nodes of links in L' are $\gamma_1 d_i$-separated, then both L' and $\overleftarrow{L'}$ are independent sets and $\forall l \in L'$, $A(L' \setminus \{l\}, l) < 1 - \psi$.*

Proposition 10.2 can be easily proved with the widely known technique to partition the plane into rings [55, 114, 137] by noticing that for any link $l \in L'$, the distance from any node in $X(L') \setminus \{l\})$ to $r(l)$ is at least $\gamma_1 d_i - d_i$.

Lemma 10.3. *For each S_i in Algorithm 7: (i) the set of sender nodes of links in S_i is an (ω_1, ω_2)-ruling of $\{x(k) : x(k) \in X_i$ and $r(k) \in R_i, \forall k \in L_i\}$, (ii) both S_i and $\overleftarrow{S_i}$ are independent sets, and (iii) $\forall l \in S_i$, $A(S_i \setminus \{l\}, l) < 1 - \psi$.*

Proof. Statement (i) directly follows from the definition of restricted ruling. Therefore, the senders of links in S_i are $\gamma_1 d_i$-separated due to the ruling property. Statements (ii) and (iii) are direct results from Proposition 10.2. □

In the ith phase of Step 2, it selects a subset S'_i of links from S_i, such that $\sum_{j \neq i} A^E(S_j, l) < \psi, \forall l \in S'_i$. Due to Lemma 10.3, $A(S_i \setminus \{l\}, l) < 1 - \psi$ for all $l \in S'_i$, and the communication on the links in S'_i from all receivers to their corresponding senders are successful. Therefore, Algorithm 7 produces a feasible solution as summarized in Lemma 10.4.

Lemma 10.4 (Correctness)**.** *The set $S' = \cup_i S'_i$ of links selected via Algorithm 7 is feasible.*

Proof. For each link $l \in S_i, \forall i$, we have (i) $A(S_i \setminus \{l\}, l) < 1 - \psi$ (in Lemma 10.3) as a result the ruling property, and (ii) $A(\sum_{j \neq i} S_j, l) < \psi$ due to Step 2 verifying the affectance. □

Estimation of affectance. At Line 7, for a link $l \in L_i$, $SP_j(l) = \sum_{l' \in S_j} \frac{P(l')}{d^\alpha(r(l'), x(l))} + N$, and $A(l, S_j) = \sum_{l' \in S_j} c_{l'} \frac{d^\alpha(l')}{P(l')} \frac{P(l)}{d^\alpha(x(l), r(l'))}$. Since $x(l)$ knows $l' \in L_j$, it can estimate $P(l')$ and $d(l')$ within a constant factor. Therefore, $x(l)$ can obtain estimated affectance $A^E(l, S_j)$ such that $A^E(l, S_j) \geq A(l, S_j)$ and is within a constant factor of $A(l, S_j)$. Likewise, at Lines 11 and 22 $r(l)$ can calculate an estimate value $A^E(S_j, l)$ of $A(S_j, l)$ with similar properties.

Simplification of Algorithm 7. The affectance checking processes in both Step 1 (Lines 3-13) and Step 2 (Lines 18-22) take $O(G(L))$ iterations in each phase i. We present it this way

Algorithm 7: Distributed MAXLSP

input : Set L of links
output: One-shot Schedule S'

/* Each $l \in L$ participates as follows */
1 $S_i \leftarrow \emptyset, S'_i \leftarrow \emptyset, X_i \leftarrow \emptyset, R_i \leftarrow \emptyset,\ \forall i = 1\ldots, g(L)$;
/* Step 1 (Lines 2-16): affectance & ruling */
2 **foreach** $i = 1, 2, \ldots, g(L)$ **do** /* $g(L)$ phases */
    /* checking affectance: Lines 3-13 */
3   **foreach** $j = 1, 2, \ldots, i-1$ **do** /* $2(i-1)$ slots */
        /* in one time slot */
4   **if** $l \in S_j$ **then** $r(l)$ transmits;
5   **if** $l \in L_i$ **then**
6             $x(l)$ senses, $SP_j(l) \leftarrow$ power sensed by $x(l)$;
7             $A^E(l, S_j) \leftarrow$ estimated $A(l, S_j)$ from $SP_j(l)$;
        /* in a second time slot */
8   **if** $l \in S_j$ **then** $x(l)$ transmits;
9   **if** $l \in L_i$ **then**
10             $r(l)$ senses, $SP'_j(l) \leftarrow$ power sensed by $r(l)$;
11             $A^E(S_j, l) \leftarrow$ estimated $A(S_j, l)$ from $SP'_j(l)$;
    /* in one time slot */
12   **if** $l \in L_i$ *and* $\sum_{j<i} A^E(l, S_j) < 1/4$ **then** add $x(l)$ to X_i;
13   **if** $l \in L_i$ *and* $\sum_{j<i} A^E(S_j, l) < 1/4$ **then** add $r(l)$ to R_i;
    /* restricted ruling: Line 15 */
14     $\omega_1 \leftarrow \gamma_1 d_i, \omega_2 \leftarrow \gamma_2 d_i$;
15     $R_i^r \leftarrow$ RESTRICTEDRULING$(\omega_1, \omega_2, X_i, R_i, \rho\gamma_1)$;
16     $S_i \leftarrow \{k : k\text{'s sender } x(k) \in R_i^r\}$;
/* Step 2 (Lines 17-24): verification */
17 **foreach** $i = 1, 2, \ldots, g(L)$ **do** /* $g(L)$ phases */
    /* checking affectance: Lines 3-13 */
18   **foreach** $j = i+1, \ldots, g(L)$ **do** /* $g(L) - i$ slots */
        /* in one time slot */
19   **if** $l \in S_j$ **then** $x(l)$ transmits;
20   **if** $l \in L_i$ **then**
21             $r(l)$ senses, $SP'_j(l) \leftarrow$ power sensed by $r(l)$;
22             $A^E(S_j, l) \leftarrow$ estimated $A(S_j, l)$ from $SP'_j(l)$;
    /* in one time slot */
23   **if** $l \in L_i$ *and* $\sum_{j \neq i} A^E(S_j, l) < \psi$ **then** $r(l)$ transmits CONFIRM message to $x(l)$;
24   **if** $l \in L_i$ *and $x(l)$ receives CONFIRM from $r(l)$* **then** add l to S'_i;
25 **return** $\cup_i S'_i$

for the ease of understanding. Now we describe how these can be simplified to $O(1)$-time operations in each phase i. At Line 16, after obtaining S_i, we immediately make all the receivers and senders of the links in S_i transmit separately in two slots, and all the senders and receivers of higher link classes senses the power and calculate $A^E(l, S_i)$ and $A^E(S_i, l)$. Then, the $O(G(L))$ iterations at Lines 4 through 11 can be omitted. In the same way, the $O(G(L))$ iterations in each phase i of Step 2 can be avoided. Lemma 10.5 summarizes the running time. The $\rho \log n$ term is from calling RESTRICTEDRULING$(\omega_1, \omega_2, X_i, R_i, \rho\gamma_1)$ (due to Theorem 10.7).

Lemma 10.5 (Running Time)**.** *Running time of Algorithm 7 after simplification is $O(G(L)\rho \log n)$ for the case of full-duplex. The restriction of half-duplex communication increases the running time by a $O(\log n)$ factor.*

Theorem 10.6 (Constant Approx.)**.** *The size of the set produced by Algorithm 7 is within a constant factor of $|OPT(L)|$.*

Proof. We prove the statement in two steps by (i) showing that $|S'|$ is within a constant factor of $|S|$, and (ii) showing that $|S|$ is within a constant factor of $|OPT(L)|$.

For the first step, we construct a $|S| \times |S|$ matrix where each element take the value of the affectance from one link to another in S, and the affectance from links in the same link class is set to zero. The sum of all the elements in the matrix is at most $|S|/2$. By Claim 2.1 in [53], $|S'| \geq |S|(1 - \frac{1}{2\psi})$.

The second step can be proved by using the Red-Blue Lemma. Let $\delta' = (3^{\alpha+1}2)$. By Claim 2.1 in [53], there exists a set $O \subseteq OPT_{2\delta'}(L)$, such that $|O| \geq |OPT_{2\delta'}(L)|/2$ and $\forall l \in O, A(l, O \backslash \{l\}) \leq \frac{1}{\delta'}$, where $OPT_{2\delta'}(L)$ denotes a maximum $2\delta'$-signal subset in L. Using the Red-Blue Lemma 10.1 with $RED = S$ and $BLUE = O \setminus S$, suppose that $|S| < |O|/5$, then there exists a link $l \in O \setminus S$, such that

$$\begin{aligned}
&A(S^-(l), l) + A(l, S^-(l)) \\
\leq &3^\alpha (A(O \setminus S, l) + A(l, O \setminus S)) \\
\leq &3^\alpha (\frac{1}{\delta'} + \frac{1}{2\delta'}) \leq \frac{1}{4}.
\end{aligned}$$

W.l.o.g., assume $l \in L_i$. The above inequality suggests that l must have passed the affectance constraint check but got removed because its sender node $x(l)$ is within at most a distance

$\gamma_2 d_i$ of the sender node $x(l')$ of some selected link l' in S_i during the restricted ruling construction (*i.e.*, Line 21 of Algorithm 8). Let L'' be the set of all the links such as l which are removed during the restricted ruling construction and are within at most a distance $\gamma_2 d_i$ of the sender node $x(l')$. Recall that γ_2 is a constant. We argue that $|L''|$ can be at most a constant: because O is a $2\delta'$-signal set, for any circular area with a radius of $\gamma_2 d_i$, there can be at most a constant number c' of links in $O \cap L_i$ whose sender nodes lie in it. Therefore, $|S| \geq |O|/5 - c'|S|$. That leads to $|S| = \Omega(|O|) = \Omega(|OPT_{2\delta'}(L)|) = \Omega(|OPT(L)|)$. □

10.4 Distributed Algorithm: Restricted (ω_1, ω_2)-Ruling

We now present RestrictedRuling$(\omega_1, \omega_2, X, R, b_{max})$ in Algorithm 8, a distributed algorithm to compute a R-restricted (ω_1, ω_2)-ruling of X, for full duplex communication under the physical interference model; in the end of the section, we extend it to the half duplex setting (where a node can perform transmission and reception/sensing at the same time) with added running time. Here, X is an arbitrary subset of senders and R is an arbitrary subset of receivers of the links in the same link class. For the algorithm to function properly, we require the input parameter $\omega_2 \geq (36\frac{\alpha-1}{\alpha-2})^{\alpha-2}\omega_1$. Recall that $B(v, d)$ denotes the ball centered at v with a radius of d. Let n be the total number of nodes. The last input parameter b_{max} denote the estimate of the maximum number of nodes in the ball $B(v, \omega_1)$ of any node $v \in X$, in the worst case, $b_{max} \leq n$.

All the nodes participating the algorithm use the same transmission power, denoted by P'. This requirement can be relaxed so that nodes may choose power levels varying within a constant factor, by modifying ω_1, ω_2 with a constant factor. We define $Thres(d) = P'/d^\alpha + N$ as a function of distance d, such that for a node v, if any other node is transmitting in a range of d, its sensed power will exceed $Thres(d)$.

In this algorithm, we call an iteration of the outer loop (Line 1) a *phase*; we call an iteration of the inner loop (Line 2) a *round*, consisting of *coordination step* (Lines 4 through 9), *confirmation step* (Lines 10 through 14) and *decision step* (Lines 15 through 21). A node v is said to be *active* if v has not joined either $\hat{Y}$ or $\hat{Z}$; otherwise, v becomes *inactive*.

In each round, the coordination step provides a probabilistic mechanism for active nodes in X to compete to get in the ruling (at Line 6). Lines 7 through 9 constitute a module to resolve the issue of sensing and transmitting at the same time, such that two nearby nodes

Algorithm 8: RestrictedRuling($\omega_1, \omega_2, X, R, b_{max}$): Distributed algorithm for computing an R-restricted (ω_1, ω_2)-ruling of X with full duplex radios.

input : $\omega_1, \omega_2, X, R, b_{max}$
output: $\hat{Y}$: an R-restricted (ω_1, ω_2)-ruling of X

`/* Each` $v \in X \cup R$ `participates as follows */`

1 **for** $i_{out} = 1$ *to* b_{max} **do** `/*` $O(b_{max})$ `phases */`
2   **for** $i_{in} = 1$ *to* $C_4 \log n$ **do** `/*` $O(\log n)$ `rounds */`
3   **if** *v is active* **then**
      `/* Coordination Step (Lines 4-9): 1 slot */`
4       $U(v) \leftarrow 0$;
5   **if** $v \in X$ **then**
6         $U(v)$ flips to 1 w/ prob. $\frac{1}{b_{max} - i_{out} + 2}$;
7   **if** $U(v) = 1$ **then**
8   v transmits and senses, $SP(v) \leftarrow$ the power v receives in this slot;
9   **if** $SP(v) > Thres(\omega_1, P(v))$ **then** $U(v) \leftarrow 0$;
      `/* Confirmation Step (Lines 10-14): 2 slots */`
10   **if** $U(v) = 1$ **then**
11   v transmits REQUEST containing its own and the receiver's id's;
12   **else if** *$v \in R$ successfully receives a REQUEST* **then**
13   **if** *REQUEST is from its corresponding sender* **then**
14   v transmits ACK;
      `/* Decision Step (Lines 15-21): 1 slot */`
15   **if** $U(v) = 1$ **then**
16   **if** *v successfully receives an ACK* **then**
17   v transmits CONFIRM, v joins $\hat{Y}$; `/*` v `becomes inactive */`
18   **else** v joins $\hat{Z}$; `/*` v `becomes inactive */`
19   **else if** $v \in X$ **then**
20   v senses, $SP(v) \leftarrow$ the power v receives in this slot;
21   **if** $SP(v) > Thres(\omega_1)$ **then** v joins $\hat{Z}$; `/*` v `becomes inactive */`

22 **return** $\hat{Y}$;

do not both enter the ruling (*i.e.*, Lemma 10.9). Next, during the decision step, a subset of active nodes decide to join $\hat{Y}$ or $\hat{Z}$.

Before a node $v \in X$ that survived the coordination step with $U(v) = 1$ notify all the nodes in the neighborhood, the confirmation step (Lines 10 through 14) asks v's corresponding receiver node to "approve" its request to be included in the ruling. That is, v survives the confirmation step if and only if

(1) $U(v) = 1$ after the coordination step;
(2) the corresponding receiver v' of v appears in R;
(3) v' successfully receives its REQUEST message; and
(4) v successfully receives ACK from v'.

The sensing operation at line 9 ensures that the set of nodes that satisfy the first two conditions above will be at least ω_1-sepatated, so that conditions (3) and (4) above will be satisfied.

In each phase, there are $C_4 \log n$ rounds, such that we can ensure a fraction of the node population have either joined $\hat{Y}$ or $\hat{Z}$, and we expect the maximum number of active nodes in the nearby region of any active node to decrease by at least 1. After each phase, the probability for each active node to access the channel and compete increments by $1/b_{max}$ (at Line 6). After the total of b_{max} phases, we have Lemmas 10.8, 10.10, 10.11 that lead to Theorem 10.7.

Theorem 10.7 (Correctness)**.** *Algorithm 8 terminates in $O(b_{max} \log n)$ time. By the end of the algorithm: (1) $\hat{Y}$ forms an R-restricted (ω_1, ω_2)-ruling of X and (2) $\hat{Z} = X \setminus \hat{Y}$ w.h.p.*

Theorem 10.7 follows directly from the lemmas below. Lemmas 10.8, 10.9 and 10.11 prove that $\hat{Y}$ is an R-restricted (ω_1, ω_2)-ruling of X, w.h.p. Lemmas 10.8, 10.10, 10.11 together shows that $\hat{Z}$ complements $\hat{Y}$ in X, and all nodes in $\hat{Z}$ such that their corresponding receivers appear in R are ω_2-covered by $\hat{Y}$ w.h.p. To help the reading flow, we defer much of the technical content — the proof of Lemma 10.8 (which involves Lemmas 10.14, 10.15 and 10.16), Lemma 10.11 and Lemma10.12 — to Appendix.

Lemma 10.8 (Completion)**.** *By the end of the algorithm, all nodes in X have joined either $\hat{Y}$ or $\hat{Z}$,* i.e.*, all nodes in X become inactive, w.h.p.*

Lemma 10.8 implies that $\hat{Z} = X \setminus \hat{Y}$. We say a node $v \in \hat{Y}$ is "*good*," if and only if $d(v, v') \geq \omega_1, \forall v' \in \hat{Y}$ and $v' \neq v$. In Algorithm 8, When a node enters $\hat{Y}$, it makes sure that there are no other ones entering $\hat{Y}$ within a range of ω_1, and it deactivate all the active nodes in the same range. Therefore, we have the following Lemmas 10.9 and 10.10.

Lemma 10.9 (Quality of $\hat{Y}$)**.** *All nodes in $\hat{Y}$ are good.*

Lemma 10.10 (Quality of $\hat{Z}$: Part 1)**.** *$\hat{Z}$ contains all the nodes ω_1-covered by $\hat{Y}$.*

Lemma 10.11 (Quality of $\hat{Z}$: Part 2)**.** *Further, suppose all nodes in $\hat{Y}$ are good, then all nodes in $\hat{Z}$ such that their corresponding receivers appear in R are ω_2-covered by $\hat{Y}$, $\forall \omega_2 \geq (36\frac{\alpha-1}{\alpha-2})^{\frac{1}{\alpha-2}}\omega_1$.*

Half Duplex Communication. Now, we assume that nodes are in the half duplex mode, so that they cannot perform transmission and reception/sensing at the same time. In Algorithm 8, Lines 7 through 9 make use of the full duplex capability, such that Lemma 10.9 is true. To account for the case of half duplex, if we replace the one-slot deterministic full duplex mechanism (Lines 7 through 9) with a randomized $O(\log n)$-time loop — illustrated by the following lines of pseudo code — we have Lemma 10.12 for half duplex communication as the counterpart of Lemma 10.9 for full duplex. The cost incurred includes (i) the increase in the total running time to obtain an (ω_1, ω_2)-ruling by $O(\log n)$, and (ii) a weakened statement in Lemma 10.12 compared to Lemma 10.9.

In replacement of Lines 7 through 9 in Algorithm 8 for using half duplex radios.

for $j = 1$ *to* $C_5 \log n$ **do** `/* resolving half duplex communication */`
    `/* in each slot */`
    if $U(v) = 1$ **then** v transmits with prob. 1/2;
    if v *does not transmit* **then**
        v senses, $SP(v) \leftarrow$ the power v receives in this slot;
        if $SP(v) > Thres(\omega_1)$ **then** $U(v) \leftarrow 0$; `/* stops */`

Lemma 10.12 (Quality of $\hat{Y}$: Half Duplex Mode)**.** *All nodes in $\hat{Y}$ are good, w.h.p.*

Since Lemmas 10.8, 10.10 and 10.11 remain valid, we obtain the following theorem for the half duplex case.

Theorem 10.13 (Half Duplex)**.** *There exists a modified version of* RestrictedRuling$(\omega_1, \omega_2, X, R, b_{max})$ *for the half duplex case, such that it finishes in $O(b_{max} \log^2 m)$ time and by the end of the algorithm: (1) $\hat{Y}$ forms an R-restricted (ω_1, ω_2)-ruling of X and (2) $\hat{Z} = X \setminus \hat{Y}$ w.h.p.*

10.5 Simulation Results

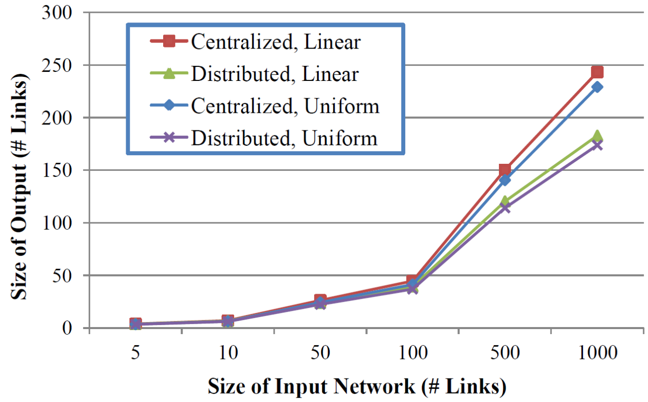
We study the empirical performance of our distributed algorithms here. We study the approximation guarantee and different quantities associated with our distributed algorithms. Our main observations are (i) the approximation ratio of our algorithm is close (within a factor of 2) of the sequential algorithm, and (ii) the topology, especially the sizes of the different length classes impacts the performance, and the links which are removed based on different criteria in the algorithm. We expect these insights can be useful in improving the empirical performance of our distributed algorithms.

We used randomly generated network topologies with 5, 10, 50, 100, 500, 1000 links, with the links spanning a total of 8 length classes (corresponding to a range of link lengths from 0.01 to 1.4). We considered linear and uniform power assignment schemes, as they represent the two extremes of sublinear power assignments. A simulation scenario specifies the network size, the link classes involved, the scheduling algorithm and the power assignment in use. We fix the parameters in the algorithms for all simulation scenarios, and average over 500 iterations.

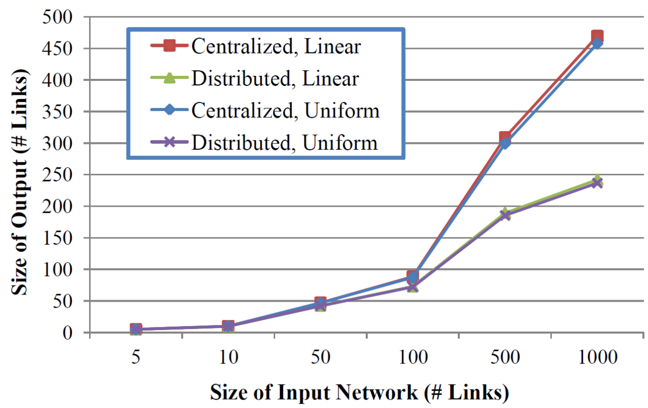
Figure 10.1 shows the gap between the output sizes of the distributed and centralized algorithms. For topologies with 3 small length classes, the gap is close to 2. For topologies with up to 8 length classes, or with only 3 large length classes, the gap is even smaller. Figure 10.2 illustrates the number of selected links, the number of removed links due to affectance violation, and the number of removed links due to ruling construction in different scenarios for the distributed algorithm. The mechanism taking the most effect in each scenario can differ drastically, *e.g.*, in Figure 10.2a 60% of the links were removed due to affectance violation, whereas in Figure 10.2a over 90% of the links were removed due to the ruling construction.

10.6 Chapter Summary

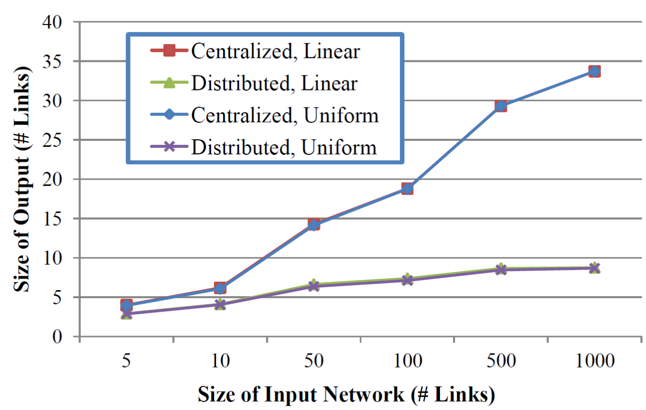
In this chapter, we present the first set of fast distributed algorithms in the SINR model for the MaxLSP problem with non-uniform power assignment. Our algorithms give constant factor approximation guarantees, matching the bounds of the sequential algorithms. Our algorithm is randomized and crucially relies on physical carrier sensing for the distributed communication steps. We find that the specific wireless device capability of duplex/half-



(a) Network consisting of 8 link classes.

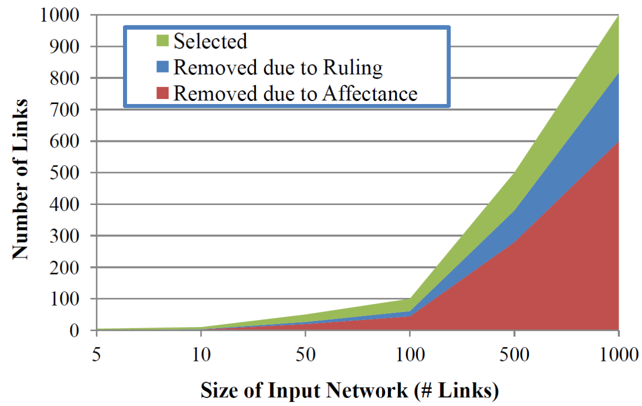


(b) Network consisting of 3 small link classes.

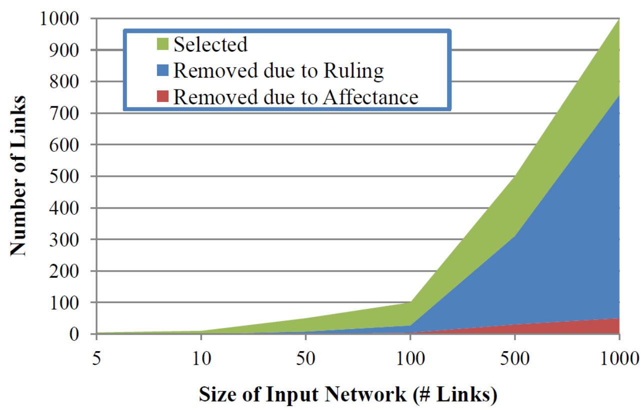


(c) Network consisting of 3 large link classes.

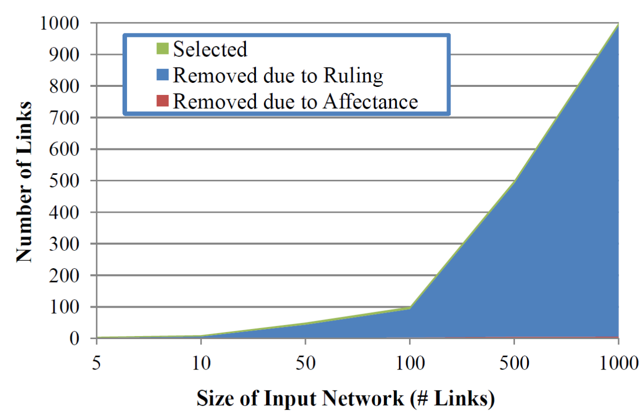
Figure 10.1: Performance of our distributed algorithm comparing to the centralized algorithm in [53]. In the legends, "Centralized" represents the centralized algorithm, "Distributed" represents the distributed algorithm, "Linear" represents linear power assignment, and "Uniform" represents uniform power assignment.



(a) Network consisting of 8 link classes.



(b) Network consisting of 3 small link classes.



(c) Network consisting of 3 large link classes.

Figure 10.2: Profiling of our distributed algorithm in terms of number of selected links, number of removed links due to affectance violation, and number of removed links due to ruling construction.

duplex communication significantly impacts the performance. Our main technique is based on the distributed estimation of affectance and restricted ruling computation, which are likely to be useful in the design of other distributed algorithms.

10.7 Appendix to Section 10.4

Recall that we call an iteration of the outer loop (Line 1) a phase of the algorithm, and an iteration of the inner loop (Line 2) a round; recall that $B(v, d)$ denotes the ball centered at v with a radius of d. Let $A_t^{W_1}(v, d)$ denote the set of *active* nodes in set W_1 that fall in the ball $B(v, d)$ at time point t; we will explicitly point t out whenever we use $A_t^{W_1}(v, d)$.

10.7.1 Proof of Lemma 10.8

The following definitions are only involved in Lemmas 10.14 and 10.15. Let η be a constant $> \left(96\frac{\alpha-1}{\alpha-2}\right)^{1/\alpha}$. In one round (which corresponds to one iteration of the inner loop) of Algorithm 8, let $U \in W_1$ be the set of nodes with $U() = 1$ at line 6 in the coordination step. We say an active node $v \in W_1$ is "lucky" in a round, with t_0 being the time that the round starts, if and only if

(1) $v \in U$;

(2) $U \cap A_{t_0}^{W_1}(v, \eta\omega_1) = \{v\}$, *i.e.*, v has no nearby active nodes in U;

(3) $SP(U \setminus A_{t_0}^{W_1}(v, \eta\omega_1), v) < Thres(\omega_1)$, *i.e.*, total power received from faraway active nodes is small.

In a round if v gets lucky, $U(v)$ will remain 1 till the end of that round, and thus will elect to be included in $\hat{Y}$ and will cause all nodes in $A_{t_0}^{W_1}(v, \omega_1)$ to get into $\hat{Z}$.

Lemma 10.14. *In a round i_{in} of phase i_{out}, with t_1 being the time that the round starts, suppose that for each active node $u \in W_1$, $|A_{t_1}^{W_1}(u, \omega_1)| \leq b_{max} - i_{out} + 1$ at the beginning of the round, then the probability for an arbitrary active node in W_1 to be lucky in the round is at least $\frac{(1/4)^{(2\eta+1)^2}/2}{b_{max}-i_{out}+2}$.*

Proof. For a round at phase i_{out}, we prove the statement in the following four steps.

(1) First, for any active node $v \in W_1$, the probability for v to be in U is $Prob\big(U(v) = 1\big) = \frac{1}{b_{max} - i_{out} + 2}$.

(2) Second, for any active node $v \in W_1$, due to the packing property we upper-bound the size of $A_{t_1}^{W_1}(v, \eta\omega_1)$ as

$$
\begin{aligned}
\big|A_{t_1}^{W_1}(v, \eta\omega_1)\big| &\le \frac{\pi(\eta\omega_1 + \omega_1/2)^2}{\pi(\omega_1/2)^2} \max_{v'} \big|A_{t_1}^{W_1}(v', \omega_1)\big| \\
&\le (2\eta + 1)^2 (b_{max} - i_{out} + 1).
\end{aligned}
$$

Then, because $b_{max} > i_{out}$, $\frac{1}{b_{max} - i_{out} + 2} \le 1/2$. the probability for all nodes other than v in $A_{t_1}^{W_1}(v, \eta\omega_1)$ to not appear in U (*i.e.*. to remain silent) is

$$
\begin{aligned}
& Prob\big(A_{t_1}^{W_1}(v, \eta\omega_1) \cap U \setminus \{v\} = \emptyset\big) \\
\ge & \prod_{u \in A_{t_1}^{W_1}(v, \eta\omega_1)} \big(1 - Prob(U(u) = 1)\big) \\
\ge & \big(1 - \frac{1}{b_{max} - i_{out} + 2}\big)^{|A_{t_1}^{W_1}(v, \eta\omega_1)|} \\
\ge & \big(1 - \frac{1}{b_{max} - i_{out} + 2}\big)^{(2\eta+1)^2 (b_{max} - i_{out} + 1)} \\
\ge & (1/4)^{(2\eta+1)^2/2}.
\end{aligned}
$$

(3) Third, for any active node $v \in W_1$, we lower-bound the probability that v's received power $SP(U \setminus A_{t_1}^{W_1}(v, \eta\omega_1), v)$ from outside of the ball $B(v, \eta\omega_1)$ is "low" — *i.e.*, below $Thres(\omega_1)$ — by (i) partitioning the plane into concentric rings via a similar technique to that in [25, 103], and (ii) referring to an $(\eta\omega_1, \eta\omega_1)$-ruling and determining the expected number of nodes in each a ring that appear in U, so that we can bound the power received from all the nodes in the rings outside of $B(v, \eta\omega_1)$.

We partition the plane into rings all centered at v, each of width $\eta\omega_1$. Let $Ring(h)$ denote the hth ring, which contains every node v' that satisfies $h\eta\omega_1 \le d(v', r(l)) < (h + 1)\eta\omega_1$, for each $h = 1, 2, \ldots$; let $Ring^a(h)$ denote the set of active nodes in $Ring(h)$. When $h = 0$, $Ring(0)$ corresponds to the ball $B(v, \eta\omega_1)$. Let $R(h)$ denote an $(\eta\omega_1, \eta\omega_1)$-ruling of $Ring^a(h)$. Then by noticing that (i) $Ring^a(h) \subseteq \cup_{v' \in R(h)} A_{t_1}^{W_1}(v', \eta\omega_1)$, and (ii) for any two nodes $v', u' \in$

$R(h), d(v', u') > \eta\omega_1$, we have

$$\begin{aligned}
&\mathbb{E}\left\{\left|U \cap Ring^a(h)\right|\right\} = \sum_{v' \in Ring^a(h)} \mathbb{E}\{U(v')\} \\
&\leq \sum_{v' \in R(h)} \sum_{u' \in A_{t_1}^{W_1}(v', \eta\omega_1)} \mathbb{E}\{U(u')\} \\
&= \sum_{v' \in R(h)} \frac{\left|A_{t_1}^{W_1}(v', \eta\omega_1)\right|}{b_{max} - i_{out} + 2} \leq 2\left|R(h)\right|.
\end{aligned} \tag{10.1}$$

To bound the cardinality of $R(h)$, we use the following facts: (i) for any two nodes $v', u' \in R(h)$, two disk centered at v', u' respectively with a radius of $\eta\omega_1/2$ are non-overlapping; and (ii) For any node $v' \in Ring^a(h)$, such a disk is fully contained in an extended ring $Ring'(h)$ of $Ring(h)$, with an extra width of $\eta\omega_1/2$ at each side of $Ring(h)$. Then, by referring to the ratio between the areas of $Ring'(h)$ and a disk, we have $\left|R(h)\right| \leq 8(2h+1)$; Inequality (10.1) yields $\mathbb{E}\left\{\left|U \cap Ring^a(h)\right|\right\} \leq 2^4(2h+1)$.

Therefore, v's received power from outside of the ball $B(v, \eta\omega_1)$ is

$$\begin{aligned}
&\mathbb{E}\{SP(U \setminus A_{t_1}^{W_1}(v, \eta\omega_1), v)\} \\
=&\mathbb{E}\Big\{\sum_{h=1}^{\infty} \sum_{v' \in U(h)} \frac{P'}{d^\alpha(v', v)} + N\Big\} \\
\leq&\sum_{h=1}^{\infty} \mathbb{E}\left\{\left|U \cap Ring^a(h)\right|\right\} \frac{P'}{(h\eta\omega_1)^\alpha} + N \\
\leq&\sum_{h=1}^{\infty} \frac{2^4(2h+1)}{h^\alpha} \frac{P'}{(\eta\omega_1)^\alpha} + N \\
\leq&\frac{48}{\eta^\alpha} \frac{\alpha - 1}{\alpha - 2} \frac{P'}{\omega_1^\alpha} + N \leq Thres(\omega_1)/2.
\end{aligned}$$

According to Markov's Inequality,

$$Prob\big(SP(U \setminus A_{t_1}^{W_1}(v, \omega_2), v) \geq Thres(\omega_1)\big) \leq 1/2,$$

implying

$$Prob\big(SP(U \setminus A_{t_1}^{W_1}(v, \omega_2), v) < Thres(\omega_1)\big) \geq 1/2.$$

(4) Finally, combining the above three, the probability that v is lucky is at least $\frac{(1/4)^{(2\eta+1)^2}/2}{b_{max}-i_{out}+2}$.

□

Lemma 10.15. *In a round i_{in} of phase i_{out}, with t_1 being the time that the round starts, suppose that for each active node $u \in W_1$, $\left|A_{t_1}^{W_1}(u,\omega_1)\right| \leq b_{max} - i_{out} + 1$; then, for an arbitrary active node $v \in W_1$ with $\left|A_{t_1}^{W_1}(v,\omega_1)\right| \geq b_{max} - i_{out}$, the probability that at least one node in $A_{t_1}^{W_1}(v,\omega_1)$ becomes inactive by the end of the round is at least a constant C_6, where $0 < C_6 = 4^{-(2\eta+1)^2}/2 < 1$.*

Proof. In a round of phase i_{out}, a sufficient condition for one node in $A_{t_1}^{W_1}(v,\omega_1)$ to be inactive by the end of the round is that any node $u \in A_{t_1}^{W_1}(v,\omega_1)$ (where u can be v) enters the confirmation step with $U(u) = 1$, so that no matter whether u receives an ACK or not in the confirmation step at least one node in $A_{t_1}^{W_1}(v,\omega_1)$ will the algorithm. Further, that either v or any node in $A_{t_1}^{W_1}(v,\omega_1)$ gets lucky satisfies this condition. Therefore, the probability for at least one node in $A_{t_1}^{W_1}(v,\omega_1)$ to become inactive in the round is at least

$$\sum_{v' \in A_{t_1}^{W_1}(v,\omega_1)} Prob(v' \text{ is lucky}) \geq \left|A_{t_1}^{W_1}(v,\omega_1)\right| \frac{(1/4)^{(2\eta+1)^2}/2}{b_{max} - i_{out} + 2} \geq 4^{-(2\eta+1)^2}/2.$$

□

Lemma 10.16. *Let $E_{i_{out}}$ denote the event that at the end of the phase i_{out}, $\left|A_{t_1}^{W_1}(u,\omega_1)\right| \leq b_{max} - i_{out}$ for every active node $u \in W_1$, where t_1 is the time that the last round in i_{out} ends. $\forall i_{out}$, $Prob(E_{i_{out}}) \geq 1 - i_{out}/n^{C_7}$, for some positive constant C_7.*

proof by induction. At the end of the phase $i_{out} = 0$, this is trivial. Suppose that for $i_{out} = i \geq 1$ the statement is true; we show that it still holds for $i_{out} = i + 1$. At the beginning of phase $i + 1$, if we already have event E_{i+1}, we are done; otherwise, let v denote an active node such that $\left|A_{t_2}^{W_1}(v,\omega_1)\right| > b_{max} - (i+1)$, where t_2 is the time that the first round in $i+1$ starts. We show $Prob(E_{i+1}) \geq 1 - \frac{i+1}{n^{C_7}}$ as below:

(1) We choose a constant $C_4 \geq \frac{C_7+1}{\log \frac{1}{1-C_6}}$ for the inner loop at Line 2 of Algorithm 8.

(2) Under the induction assumption for phase i, we have: for any round i_{in} during phase $i + 1$, with t_3 being the time that this round starts, if $\left|A_{t_3}^{W_1}(v,\omega_1)\right| \leq b_{max} - (i + 1)$, we are done; otherwise, as long as $b_{max} - i \geq \left|A_{t_3}^{W_1}(v,\omega_1)\right| \geq b_{max} - (i + 1)$, the probability for at least one node in $A_{t_3}^{W_1}(v,\omega_1)$ to turn inactive during the round is at least C_6 according

to Lemma 10.15. Then, the probability for v to become inactive by the end of phase $i+1$ (consisting of $C_4 \log n$ rounds) is at least $1-(1-C_6)^{C_4 \log n} \geq 1-1/n^{C_7+1}$.

(3) By considering both the conditional probability and the fact that there are at most n such nodes as v,

$$\begin{aligned}
&Prob(E_{i+1})\\
\geq &Prob(E_{i+1} \mid E_i)Prob(E_i)\\
\geq &(1-\frac{n}{n^{C_7+1}})(1-\frac{i}{n^{C_7}}) \geq 1-\frac{i+1}{n^{C_7}}. \qquad \square
\end{aligned}$$

At the end of phase $i_{out} = b_{max}$, with t_4 being the time that the last round in phase i_{out} ends, we have that with probability of at least $1-\frac{b_{max}+1}{n^{C_7}}$, for every active node $u \in W_1$, $|A_{t_4}^{W_1}(u,\omega_1)| < 1$. That means that all the nodes in W_1 have joined either $\hat{Y}$ or $\hat{Z}$, with probability of at least $1-\frac{1}{n^{C_7-1}}$, concluding the proof of Lemma 10.8.

10.7.2 Proof of Lemma 10.11

Proof by contradiction. Suppose there exists a node $v \in \hat{Z}$ that is not ω_2-covered by $\hat{Y}$. Therefore, all the nodes in $\hat{Y}$ are outside of the ball $B(v,\omega_2)$. We calculate $SP(\hat{Y},v)$ and derive the conflict. We notice the following three facts: (1) all the nodes in $\hat{Y}$ have a mutual distance of at least ω_1; (2) the distance from any sender node in $\hat{Y}$ to v is $> \omega_2$; and (3) $\omega_2 > \omega_1/2 > 0$.

Proposition 10.17. *$\forall V' \in V$ and $\forall v \notin V'$, if (1) all the nodes in V' are at least ρ_1 away from each other, (2) the distance between v and any node in V' is at least ρ_2, and (3) $\rho_2 > \rho_1/2 > 0$, then $SP(V',v) < \frac{36(\alpha-1)}{\alpha-2}\frac{\rho_2^2}{\rho_1^2}\frac{P}{\rho_2^\alpha} + N$.*

Due to Proposition 10.17,

$$SP(\hat{Y},v) \leq 36\frac{\alpha-1}{\alpha-2}\frac{P}{\omega_1^2(\omega_2)^{\alpha-2}} + N \leq 36\frac{\alpha-1}{\alpha-2}\frac{\omega_1^{\alpha-2}}{\omega_2^{\alpha-2}}Thres(\omega_1) \leq Thres(\omega_1).$$

According to the condition for v to enter $\hat{Z}$, we must have $SP(\hat{Y},v) > Thres(\omega_1)$, where lies the contradiction.

10.7.3 Proof of Lemma 10.12

Suppose there is "bad" node $v \in \hat{Y}$ such that there exists a node $v' \in \hat{Y}$ and $d(v, v') < \omega_1$. We call such a v' a "bad" partner of v. The only possible situation for v to have a "bad" partner v' is that v and v' enter $\hat{Y}$ in the same round; otherwise, one of them should have been "pushed" into $\hat{Z}$ during the decision step of a round when the other enters $\hat{Y}$.

Let u denote an arbitrary node in $\hat{Y}$. The necessary and sufficient condition for u to be bad (or to have a bad partner) is that, in the coordination step of the round when u enters $\hat{Y}$, there exists at least one active node $u' \in W_1$ such that, (1) $d(u, u') < \omega_1$, (2) u and u' are in U, and (3) u and u' made the same random binary decisions all through the $C_5 \log m$ slots of the coordination step.

W.l.o.g., we assume u enters $\hat{Y}$ at round i_{in} in phase i_{out}. Let t_1 denote the time that round i_{in} in phase i_{out} begins. The probability that u is bad equals the probability for at least one other active node $u' \in A^{W_1}_{t_1}(u, \omega_1)$ (*i.e.*, $d(v, v') < \omega_1$) to enter $\hat{Y}$ at round i_{in} in phase i_{out}.

$$Prob(u \text{ is bad}) \leq \sum_{u' \in A^{W_1}_{t_1}(u,\omega_1)} Prob(U(u') = 1) \frac{1}{2^{C_5 \log n}}$$
$$\leq \frac{1}{n^{C_5}} \frac{\left|A^{W_1}_{t_1}(u, \omega_1)\right|}{b_{max} - i_{out} + 2} \leq \frac{1}{n^{C_5}} \frac{\max_{\text{active } w \in W_1} \left|A^{W_1}_{t_1}(w, \omega_1)\right|}{b_{max} - i_{out} + 1}.$$

Further, due to Lemma 10.16,

$$Prob\big(\max_{\text{active } w \in W_1} \left|A^{W_1}_{t_1}(w, \omega_1)\right| \leq b_{max} - i_{out} + 1\big) \geq 1 - \frac{i_{out} - 1}{n^{C_7}} \geq 1 - \frac{1}{n^{C_7 - 1}}.$$

Therefore,

$$Prob(u \text{ is good}) \geq (1 - \frac{1}{n^{C_5}})(1 - \frac{1}{n^{C_7-1}}) \geq 1 - \frac{2}{n^{\min\{C_5, C_7-1\}}}.$$

Finally, since $\hat{Y}$ contains at most n nodes, the probability that there are no bad nodes in $\hat{Y}$ is at least $1 - \frac{1}{n^{\min\{C_5, C_7-1\}-1}}$.

Chapter 11

Fast Distributed Approximation Algorithms for MST under the Physical Interference Model

11.1 Preliminaries and Distributed Computing Model for MST

Inequality (8.1) defines under a certain transmission power P the *transmission range*, which is the threshold distance beyond which two nodes cannot communicate with each other, and which equals $\sqrt[\alpha]{P/(N\beta)}$. To reduce notational clutter, we will say that the transmission range r associated with a transmission power P is $r = (P/c)^{1/\alpha}$, for a constant c. Let $r_{max} = (P_{max}/c)^{1/\alpha}$ denote the maximum transmission range of any node at the maximum power level. W.l.o.g., we assume $r_{max} \leq d_{max}$ in our algorithms.

We say that a set S of nodes in the plane form a "constant density set" w.r.t. range r if there are $O(1)$ nodes within the ball $B(v, r)$ of any node v, where v is the center and r is the radius. A set $S' \subset S$ is said to be a *constant density dominating set* for S w.r.t. range r, if S' is a constant density set, and for each $v \in S$, there exists $u \in S'$ such that $v \in B(u, r)$. Given a set S of nodes and range r, we define $G_r(S) = \big(S, E = \{(u, v) : u, v \in S, d(u, v) \leq r\}\big)$ to be the graph induced by S with range r.

11.1.1 Distributed Computing Model under the SINR Model

Since the network model is purely based on nodes without existing communication links, we explicitly describe our distributed computing model under the SINR model for MST.

(1) The network is synchronized and for simplicity we assume all time slots have the same length.

(2) The graph $G_{r_{max}/c}(V)$ induced by V with range r_{max}/c is connected, *i.e.*, the graph can be connected by using a transmission power level of P_{max}/c' which is slightly less than the maximum but within a constant factor.

(3) All nodes have a common estimate of n within a n^c for some constant c.

(4) All nodes share a common estimate of $d_{min} = 1$ and d_{max}, the minimum and maximum distances between nodes. We use $\mu = \log \frac{d_{max}}{d_{min}}$ to denote the "distance diversity" (similar to the "link diversity" in [46]), which is the number of classes of similar length edges into which the set of all edges can be partitioned. It is common to assume that $\mu = O(\log n)$.

(5) As mentioned earlier, we assume nodes are equipped to be able to do adaptive power control, *i.e.*, each node v can transmit at any power level $P \in [0, P_{max}]$.

11.2 Problem Definition: The Minimum Spanning Tree Problem under the SINR Model (MST-SINR)

Given a set V of wireless nodes, the goal is to find a spanning tree T, such that the total cost of T is minimized, w.r.t. a cost function $cost(u, v) = d(u, v)$ for any pair of nodes (u, v); for a set E of edges, we define $cost(T) = \sum_{(u,v)\in E} cost(u, v)$. We focus on developing distributed approximation algorithms. We say an algorithm gives a γ-approximation factor if it constructs a spanning tree T, with $cost(T) \leq \gamma \cdot cost(MST)$, where MST represents an optimum solution. In the problem we study here, we only require the tree to be constructed implicitly — we assume we have a sink node s, and we define a *parent*, $par(v)$ for all nodes $v \neq s$, such that the set of edges $\{(v, par(v)) : v \in V, v \neq s\}$ form a spanning tree. Each node only needs to know the identity of its parent. The goal of this chapter is to design an algorithm for computing an approximate MST in the distributed computing model based on

SINR constraints (described below); we do not require the transmissions (in either direction) on all the edges in the tree to be simultaneously feasible in the SINR model.

11.3 Distributed Primitives

We discuss two primitives that are needed in our MST algorithms for local broadcasting and dominating sets. The results of [45, 125] directly provide us efficient implementations for these two problems.

11.3.1 Local Broadcasting

The *local broadcasting range* r_b is the distance up to which nodes intend to broadcast their messages. We say the local broadcasting from S to S' is successful if and only if for each node $u \in S$ with transmission power P_b, all the nodes in S' within u's local broadcasting range r_b receives the message from u. We assume we have an algorithm, $\texttt{LocalBroadcast}(S, S', r_b, P_b)$, which takes sets S, S' of nodes, a distance r_b and a power level P_b as input, and ensures that the local broadcasting from S to S' is successful. A small modification of the local broadcasting scheme of [45] achieves this step, which is described below.

Lemma 11.1. *Given two sets S, S' of nodes, a local broadcasting range r_b and a power level $P_b = c' r_b^\alpha$, for a constant c', there is a distributed algorithm $\texttt{LocalBroadcast}(S, S', r_b, P_b)$ that runs in $O(N(S, \gamma r_b) \log |S|)$ time where $N(S, \gamma r_b)$ is the maximum number of nodes in S within a distance γr_b of any node in S and γ is a constant, such that: (i) each node $v \in S'$ receives the message from all nodes in S within distance r_b w.h.p., and (ii) each node $v \in S'$ is able to selectively ignore messages from any node $u \in S'$ which is beyond distance γr_b.*

Proof(sketch). We describe the protocol as follows. Following [45] we use random access, in which each node in S transmits at Power P_b with prob. $\frac{1}{N(S,\gamma r_b)}$ (known to each node in S), and each node in S' senses the channel to receive messages. Each time a node $v \in S'$ receives a message, v checks the total power received; if that exceeds $\frac{P_b}{(\gamma' r_b)^\alpha} + N$, v discards the received message.

Property (i) in Lemma 11.1 directly follows from the proofs in [45], by partitioning the space into rings and uppering bounding the stochastic interference. Then Property (ii) follows from the condition we put on accepting a message based on $SP(v)$, such that any message sent from a node beyond distance γr_b from v will be ignored, where γ is a constant. □

11.3.2 Constant Density Dominating Set

We assume an algorithm `ConstDominatingSet`(S, r_c) that takes as input a set S of nodes and a range $r_c \leq r_{max}$, and produces a constant density dominating set $S' \subset S$ corresponding to this range, such that for each node in S there is a node in S' which is with distance r_c, and the density of the nodes in S' is at most a constant. By density, we mean the number of nodes in S' within a range r_c of any node in S'. The algorithm in [125] serves this purpose, with the performance summarized below.

Lemma 11.2 ([125]). *Given a set S of nodes and a range $r_c \leq r_{max}$, a constant density dominating set for S can be constructed in time $O(\log n)$ w.h.p. under the SINR model.*

11.3.3 Nearest Neighbor Tree Scheme

The algorithmic paradigm underlying our algorithm is the Nearest Neighbor Tree (NNT) scheme [71, 72, 73]. The NNT scheme can be used to construct a spanning tree, called the *Nearest Neighbor Tree (NNT)*, efficiently in a distributed fashion. The cost of the NNT can be shown to be within an $O(\log n)$ factor of the cost of the MST. The scheme used to construct an NNT (henceforth called *NNT scheme*) consists of the following two steps:

(1) each node first chooses a unique *rank* from a totally-ordered set; a ranking of the nodes corresponds to a permutation of the nodes;

(2) each node (except the one with the highest rank) connects (via the *shortest path*) to the *nearest* node of higher rank.

It can be shown that the NNT scheme constructs a spanning subgraph in any weighted graph whose cost is at most $O(\log n)$ times that of the MST, irrespective of how the ranks are selected (as long as they are distinct) [72, 73]. Note that some cycles can be introduced in step 2, and hence to get a spanning tree we need to remove some edges to break the cycles. The main advantage of the NNT scheme is that each node, individually, has the task of finding its nearest node of higher rank to connect to, and hence no explicit coordination is needed among the nodes.

However, despite the simplicity of the NNT scheme, it is non-trivial to efficiently implement the scheme in an arbitrary weighted graph. The work of [71] showed how to efficiently implement the scheme in an arbitrary weighted graph in the CONGEST (wired) model. This implementation was shown to have a running time of $\tilde{O}(D(G) + L(G, w))$ where $L(G, w)$ is a parameter called the *local shortest path diameter* and $D(G)$ is the (unweighted) diameter of the graph. This distributed implementation cannot be directly used here, due to the (additional) complication of having to obey SINR constraints when each node tries to search for the nearest node of higher rank to connect to. Second, the local shortest path diameter can be significantly larger than the diameter of the underlying graph. In particular, it can be as large as n, in which case, the above implementation does not give the time bound that we would like to show in this chapter, *i.e.*, close to the diameter of the underlying graph. Note that this is essentially the best possible, since computing a spanning tree on an arbitrary graph takes diameter time. To show this stronger bound, we exploit the fact that the underlying graph has a geometric structure. The main idea is to choose ranks in a particular way (in step (1) of the NNT scheme) that guarantees the each node can find its nearest node of higher rank fast. This idea was first implemented in the UDG-NNT (Unit disk graph-NNT) algorithm [73]. However, again this implementation does not directly work in the SINR model. We show that one has to implement both step (1) and step (2) of the NNT scheme in a incremental and staged fashion, so that SINR constraints are obeyed and still many nodes can progress simultaneously. We show that NNT technique results in spatial separation of "active" nodes (*i.e.*, nodes that need to communicate) in each step, and hence is amenable to obey SINR constraints. The details are in the next Section.

11.4 Approximating MST in the SINR Model

We now discuss the algorithm `MST-SINR` for finding a low cost spanning tree. We start with the main intuition for the algorithm. The basic idea for satisfying SINR constraints at each step is to ensure that the senders are "spatially separated". In many distributed MST algorithms, *e.g.*, the algorithms of [38, 40], there is no guarantee that senders are well separated. However, the *Nearest Neighbor Tree (NNT)* scheme [71, 72, 73] has a nice property that nodes gradually increase their range, and the nodes which are active in any round are sufficiently well separated. However, this approach has only been studied in CONGEST model in [71, 72, 73], and we adapt it to the SINR model. For illustration, first imagine that the maximum power P_{max} is "unbounded" so that the $d_{max} \leq r_{max}$. The algorithm in this special case involves the following steps.

(1) We have $\log r_{max} \leq \mu$ phases, ranging from $i = 1, \ldots, \log r_{max}$.
(2) In the ith phase, a subset S_i of nodes participate, and the edges chosen so far form a forest rooted at nodes in S_i. The nodes in S_i transmit at power level of $c \cdot d_i^{\alpha}$ for a constant c, where $d_i = 2^i$.
(3) In the ith phase, each node $v \in S_i$ runs the NNT scheme: v connects to a "close-by" node in S_i within distance $c' \cdot d_i$ of higher rank, if one exists, for a constant c'. The nodes which are not able to connect continue into phase $i + 1$.

We need to prove that the above steps can be implemented in the SINR model, and the resulting implicit tree is a low cost tree, relative to the MST. One complication, in contrast to the original NNT scheme, is that there is no way to ensure that a node $v \in S_i$ connects to the "closest" node in S_i of higher rank, within distance d_i. Because of the SINR model, it is possible that a transmission from some far away node could be received by v. We show that the probability of this event is very low, so that with high probability, we get a tree that is "close to" the NNT, leading to the logarithmic approximation bound for the cost of the tree. We describe the above algorithm as subroutine `NNT-SINR-BP` in Section 11.4.1, with the modification that P_{max} is bounded, and instead of μ phases, we only have $\log r_{max}$ phases. The result of `NNT-SINR-BP` is a forest with the highest rank nodes forming the roots.

In the general case, P_{max} is bounded, so that $d_{max} > r_{max}$, and all we have is that the graph induced by range r_{max} is connected*. If we run the above bottom-up algorithm with some maximum range r_1, we would get a forest in which the roots are at least $c \cdot r_1$ apart, where c is a constant. In order to connect up the roots, we first use a "top-down" approach where we choose a set Dom of nodes so that (i) each node $v \notin Dom$ is within distance r_1 of some node in Dom, and (ii) for each node $u \in Dom$, there are a "small" number of nodes within distance r_1. The idea is that if we can construct a spanning tree on Dom (in the SINR model), and can ensure that each node in $V \setminus Dom$ has a higher rank neighbor in Dom, within distance r_1, then the above bottom-up phase restricted to $V \setminus Dom$ will allow the trees to be combined. Such a set Dom is precisely a "constant density dominating set", as defined and computed efficiently in [125]. The spanning tree on Dom is constructed by adapting the UDG-NNT algorithm of [73], which starts with a node s and spreads the ranks

*Recall that in Section 11.1, we in fact assume something stronger: the graph induced by a range of r_{max}/c is connected for a constant c.

from it. This adaptation is discussed below as subroutine `NNT-SINR-CD`, which also works in the SINR model because the nodes in Dom are spatially well separated.

We first describe the two subroutines, and then discuss the main algorithm.

11.4.1 Constructing Forests with Power Constraints

Algorithm 10: `NNT-SINR-BP`$(S, P_{max}, rank(\cdot))$

Input : node set S, max power P_{max}, and value $rank(v)$ for each $v \in S$
Output: set F of edges

Each node $v \in S$ does the following:
$r_{max} \leftarrow (P_{max}/c)^{1/\alpha}$, $\mathcal{K} \leftarrow \lfloor \log_2 r_{max} \rfloor$;
initially $S_i \leftarrow \emptyset, \forall i > 1$; # S_i `denotes the set of active nodes in phase` i
$S_1 \leftarrow S$;
for $i = 1$ *to* $\mathcal{K}$ **do** # `in each phase`
  # v `broadcasts its rank to all nodes in` S_i `within range` d_i
  $d_i \leftarrow 2^i$, $P_i \leftarrow c' d_i^{\alpha}$; # c' `is a constant`
  if $v \in S_i$ **then** # `broadcast rank value`
    v broadcasts its rank by participating in `LocalBroadcast`(S_i, S_i, d_i, P_i);
  $S_v \leftarrow$ set of nodes with ranks received by v;
  $v' \leftarrow$ the node with highest rank in S_v;
  if $v = v'$ **then** # v `has highest rank locally: add` v `to the active set` S_{i+1} `for next phase`
    add v to S_{i+1};
  else $par(v) \leftarrow v'$; # v `is done`
$F \leftarrow \{(v, par(v)) : par(v) \neq \emptyset, \forall v \in S\}$;

For subroutine `NNT-SINR-BP`, we are given a set S, a maximum power level P_{max} (corresponding to a range r_{max}), and a rank function $rank(\cdot)$, which assigns unique ranks for all nodes in S. The goal is to construct a forest, in which each node connects to a parent within range r_{max}. As discussed earlier, `NNT-SINR-BP` uses the NNT approach of [71,72,73], in which each node connects to the nearest node of higher rank, which leads to a forest. However, in order to be feasible in the SINR model, we need to do this in a careful and staged manner, so that the set of transmitting nodes in each round form a constant density set.

Lemma 11.3. *At the beginning of each phase i, for each node $v \in S_i$, there are at most a constant number of active nodes in the ball $B(v, d_i)$*, i.e., $\left|B(v, d_i) \bigcap S_i\right| = O(1)$. *Further, for* *each node $v \in S_i$,there are no other active nodes in the ball $B(v, d_i/2)$*, i.e., $B(v, d_i/2) \bigcap S_i = \{v\}$.

Proof by induction. Recall that S_i is the set of active nodes at phase i. For phase 1, $S_1 = S$ and $d_0 = 1$; obviously there can be at most 16 nodes in the ball. Assume that for phase i, the statement is true. Then, at the beginning of phase $i+1$, if v is active, according to the algorithm, all nodes in the range of d_i of v should all have a lower rank and thus are all inactive. Therefore, the distance between any two nodes in S_{i+1} is at least d_i. By packing property, there can be at most 16 nodes in the ball $B(v, d_{i+1})$. Therefore, the statement is true for every phase. □

Lemma 11.4. *In the end of the algorithm, the set F forms a forest. Further, the set $S_\mathcal{K}$ forms the set of roots of F, and for any two nodes $u, v \in S_\mathcal{K}$, we have $d(u, v) \geq r_{max}/2$.*

Proof. Because of the NNT property, in which each node only connects to a higher rank parent, there are no cycles. Next, by design, the nodes in $S_\mathcal{K}$ do not connect to any other node, and form the roots of the forest F. By Lemma 11.3, it follows that for any $u, v \in S_\mathcal{K}$, we have $d(u, v) \geq r_{max}/2$. □

Lemma 11.5. *The cost of the forest F is at most $O(\mu)$ times that of of the minimum spanning forest of the nodes in F. Further,* `NNT-SINR-BP` *is feasible in the SINR model, with running time of $O(\mu \log n)$, w.h.p.*

Proof. For $i < \mathcal{K}$, let $F_i = \{(v, w) \in F : v \in S_i, w = par(v) \in S_{i+1}\}$ denote the set of edges in F between nodes in sets S_i and S_{i+1}. Recall that S_i denotes the set of nodes which remain active in the beginning of phase i. Let $MST(S_i)$ denote a minimum spanning tree of only the nodes in S_i. W.l.o.g., we assume $|S_i| > 1$. For any pair of nodes in S_i, the mutual distance is at least $d_i/2$ (from Lemma 11.3), it follows that $cost(MST(S_i)) \geq (|S_i| - 1)d_i/2 \geq |S_i|d_i/4$. Next, we have $cost(F_i) \leq d_i|F_i| \leq d_i|S_i|$, implying $cost(F_i) \leq 4cost(MST(S_i))$. Since there are $O(\log r_{max})$ phases and $cost(MST(S_i)) \leq cost(MST)$, we have $cost(F) = \sum_i cost(F_i) \leq 4(\log r_{max}) \sum_i cost(MST(S_i))$. We have assumed $r_{max} \leq d_{max}$; therefore, $\log r_{max} \leq \mu$, implying the statement of approximation ratio. By Lemmas 11.1 and 11.3, each run of the local broadcast primitive finishes in $O(\log n)$ time, so does each phase of the algorithm. The statement of running time follows. □

11.4.2 Spanning Trees for Instances with Constant Density

Algorithm 11: `NNT-SINR-CD`(S, s, r)

Input : constant density node set S, sink s, range r
Output: spanning tree T on S

1 Each node $v \in S$ does the following:
`# in phase 0:`
2 **if** $v = s$ **then** `#` s `broadcasts to neighbors`
3   v generates a large $rank(v)$, and broadcasts it by `LocalBroadcast`$(\{s\}, S, r, cr^{\alpha})$;
4 **else** v listens;
`# in each phase` i`:`
5 **for** $i = 1$ *to* $2D(G_r(S))$ **do**
6     $S_i \leftarrow \emptyset$;
7   **if** v *receives a message* $m = (id', rank')$ **then**
8   **if** m *is the first message* v *ever receives* **then**
9   v generates $rank(v)$ randomly such that $rank(v) < rank'$;
10             $S_i \leftarrow S_i \cup \{v\}$;
11   v participates in `LocalBroadcast`(S_i, S, r, cr^{α});
12   add id' to list $L(v)$, **if** $rank' > rank(v)$;
13 $par(v) \leftarrow$ one of the nodes in $L(v)$;
14 **return** the set $\{(v, par(v)) : v \neq s\}$ of edges

We now consider the problem of constructing a low cost spanning tree for a constant density instance. Subroutine `NNT-SINR-CD` takes such a set S, a sink s and a range r as input. As defined in Section 11.1, $G_r(S)$ denotes the S-induced graph based on range r. We adapt the UDG-NNT algorithm of [73] for producing a spanning tree. Let $D(G_r(S))$ denote the diameter of $G_r(S)$; we assume (an estimate of) $D(G_r(S))$ is known to all the nodes. We note that in Line 11 of the algorithm, the local broadcasts are run simultaneously for all nodes v, which get the rank message for the first time. Further, each iteration of the **for** loop in Lines 6-12 involves $O(\log n)$ time steps, in which the local broadcast is run; nodes v which do not have to send their ranks in some iteration remain silent during these steps.

Lemma 11.6. *If $G_r(S)$ is connected, and S is a constant density set with range r, Algorithm* `NNT-SINR-CD` *produces under the SINR model a spanning tree T_1 on S with $cost(T_1) = O(OPT(G_r(S)))$ in time $O(D(G_r(S)) \log n)$ with high probability, where $OPT(G_r(S)) = cost(MST(G_r(S)))$.*

Proof. Our proof mimics the proof of algorithm UDG-NNT from [73]. Since $G_r(S)$ is connected, each node $v \neq s$ is able to run line 12, and the tree T_1 is constructed. The simultaneous calls to `LocalBroadcast` all take time $O(\log n)$ in each round (with high probability), so that the overall running time is $O(D(G_r(S)) \log n)$, w.h.p.. Since the `LocalBroadcast` algorithm is feasible in the SINR model, `NNT-SINR-CD` is also feasible.

The constant density property of S implies that any ball $B(v, r')$ with $r' \leq r$ has $O(1)$ nodes in S. Therefore, $cost(MST(G_r(S))) = \Omega(|S|r)$. Next, the local broadcasts (Lemma 11.1) ensure that each node v receives messages from nodes within distance $c'r$ for a constant c'. Therefore, $cost(T_1) \leq |S|c'r$, and the lemma follows. □

11.4.3 Putting Everything Together: Algorithm `MST-SINR`

Our high level idea is to start with a constant density dominating set Dom, and run subroutine `NNT-SINR-CD` to construct a tree T_1 on Dom. We then choose and disseminate ranks suitably, and run `NNT-SINR-BP` to form a forest, which get connected to the nodes in Dom, and together form a spanning tree.

Our analysis, at a high level, involves the following steps: we first show that T_1 is a spanning tree on Dom with range $2r_{max}/c$, and has low cost. Next, we show that Algorithm `NNT-SINR-BP` with the range of r_{max} results in a forest on $V \setminus Dom$, each of whose components gets connected to some node in Dom, because of the way the ranks are chosen. Finally, we show that the combined tree produced in this manner has low cost.

Lemma 11.7. *The graph $G_{2r_{max}/c}(Dom)$ induced by Dom w.r.t. a range of $2r_{max}/c$ is connected, and has a diameter of at most $2D$.*

Proof. Suppose $G_{2r_{max}/c}(Dom)$ is not connected. Then there exists a partition $Dom = Dom_1 \cup Dom_2$, which induce disconnected components, and $\min_{(u,v)\in Dom_1 \times Dom_2}\{d(u,v)\} > 2r_{max}/c$, since the dominating set Dom constructed uses range r_{max}/c. That implies that G is connected with r_{max}/c, which is a contradiction. Therefore, $G_{2r_{max}/c}(Dom)$ is connected. □

Lemma 11.8. *Algorithm* `MST-SINR` *produces a spanning tree of cost $O(\mu)$ times the optimal in time $O(D \log n + \mu \log n)$, with high probability, in the SINR model.*

Proof. From Lemma 11.6, the call to `NNT-SINR-CD` produces a spanning tree T_1 on set Dom. Next, consider the call to `NNT-SINR-BP` in Line 12 of the algorithm. Let $V' = \{v \in V \setminus Dom :$

Algorithm 12: MST-SINR(V)

Input : node set V

Output: spanning tree T

\# use the algorithm of [125] to construct a constant density dominating set Dom w.r.t. range r_{max}/c, for a constant c

$Dom \leftarrow$ `ConstDominatingSet`$(V, r_{max}/c)$ for a constant c;

\# we adapt the UDG-NNT algorithm of [73] for constructing a spanning tree on Dom

$s \leftarrow$ a node in Dom; # sink node

$T_1 \leftarrow$ `NNT-SINR-CD`$(Dom, s, 2r_{max}/c)$;

for *each* $v \in Dom$ **do**

  $rank(v) \leftarrow$ the rank chosen in this process;

\# each $v \in Dom$ broadcasts its rank to nodes within its range

run `LocalBroadcast`$(Dom, V \setminus Dom, r_{max}/c)$ such that all $v \in Dom$ to broadcast $rank(v)$;

for *each* $v \in V \setminus Dom$ **do**

  $b(v) \leftarrow$ largest rank received by v;

  $q(v) \leftarrow$ id of the node that sent v rank $b(v)$;

  v chooses $rank(v)$ uniformly at random such that $rank(v) < b(v)$;

  $dom(v) \leftarrow q(v)$; # to connect to tree roots of forest F_1 constructed below

\# use the ranks chosen above to build a forest with tree roots in Dom to connect up with the tree T_1

$F_1 \leftarrow$ `NNT-SINR-BP`$(V \setminus Dom, P_{max}, rank(\cdot))$;

for *each* $v \in V \setminus Dom$ **do**

  if $par(v) = \emptyset$ **then** # v is a tree root in forest F_1

    $par(v) \leftarrow dom(v)$; # connect to T_1

return set $\{(v, par(v)) : par(v) \neq \emptyset\}$ of edges.

$par(v) = \phi\}$ be the set of nodes for whom the parent is not defined during this call. Each nodes in V' corresponds to a tree root of the forest constructed in this call, *i.e.*, the edges $F_1 = \{(v, par(v)) : v \in V \setminus Dom, par(v) \neq \phi\}$ (constructed during this call) form a forest, rooted at nodes in V'. In Line 11, each node in $V \setminus Dom$ registers its dominating node in T_1; in Line 15, each node $v \in V'$ uses this information to connect to T_1. Therefore, $T_1 \cup F_1 \cup \{(v, par(v)) : \forall v \in V'\}$ is a spanning tree.

From Lemma 11.6, we have $cost(T_1) = O(OPT)$. Next, from Lemma 11.5, it follows that $cost(F_1) = O(\mu \cdot OPT)$. Finally, for each $v \in V'$, $d(v, par(v)) \leq c' r_{max}/c$. Therefore, $\sum_{v \in V'} d(v, par(v)) = O(OPT)$, which implies the cost of the tree produced by `NNT-SINR-BP` is $O(\mu \cdot OPT)$.

Finally, we analyze the running time. The call to `ConstDominatingSet` in Line 1 takes $O(\log n)$ time. From Lemma 11.6, the call to `NNT-SINR-CD` takes time $O(D \log n)$, with high probability. Next, the local broadcast takes time $O(\log n)$, and the call to `NNT-SINR-BP` takes time $O(\mu \log n)$. Putting all of these together, the total running time is $O(D \log n + \mu \log n)$, with high probability. From [125], and Lemmas 11.6, 11.5 and 11.1, all the computations of the algorithm are feasible in the SINR model. □

11.4.4 Scheduling Complexity of the Spanning Tree

The *scheduling complexity* (as defined in [103]) of a set of communication requests is the minimal number t of time slots during which each link request has made at least one successful transmission. Due to the constant factor approximation algorithm by Kesselheim [69] for finding a maximum feasible set of links, any spanning tree has a scheduling complexity of $O(\log n)$ under the SINR model. Here, we use the term *distributed scheduling complexity* to describe the scheduling complexity for a set of links that needs to make transmission decisions in a distributed fashion. The spanning tree T produced by our algorithm has the following property.

Lemma 11.9. *The spanning tree T produced by Algorithm* `MST-SINR` *has a distributed scheduling complexity of $O(\mu \log n)$, regardless of the direction of the transmission requests on the edges.*

Proof by construction. Let the edges in T be oriented in arbitrary directions, so that we obtain a set L of links, which represent a set of transmission requests.

We construct a feasible schedule which can be implemented in a distributed fashion. We partition the links into length classes $1, 2, \ldots, \log r_{max}$, such that all links in length class i have a length between 2^{i-1} and 2^i. In each phase $i \in [1, \log r_{max}/c]$, links in length class i transmit with probability $1/K$ in each time step, where K is an upper-bound of the number of edges/links in length class i with an end node within range 2^{i+1} of each node. With the same proof approach as that in [45], we can prove that in $O(K \log n)$ time steps, all the links in length class i have made successful transmissions with high probability. This way, in $\log r_{max}/c]$ phases, all the links have made successful transmissions, w.h.p.

Now, we argue that K is $O(1)$ in each phase, and thus the total schedule length is $O(\mu \log n)$. Due to the construction of T in Algorithm `MST-SINR`, since Dom is already a constant density node set, we only need to analyze the edges formed during the call to `NNT-SINR-BP`$(V \setminus Dom, P_{max}, rank(\cdot))$ in Line 12 of Algorithm `MST-SINR`. In each phase of the `NNT-SINR-BP`, the lengths of edges increases upon the previous phase. In one phase during `NNT-SINR-BP`, the newly formed edges corresponds to a length class, and the nodes involved in that phase form a constant density node set, according to Lemma 11.3. That implies that the number of edges in length class i with an end node within range 2^{i+1} of each node is $O(1)$.

In terms of the distributed implementation of scheduling, each edge may remember the phase number in `NNT-SINR-BP` when they are connected up by a new edge, and participate in the phase of the same phase number of our scheduling construction. □

11.5 Chapter Summary

In this chapter, we describe the first distributed algorithm in the SINR model for approximate minimum spanning tree construction. This is the first such result for solving a "global" problem in the emerging SINR based distributed computing model— this is in contrast to "local" problems, such as independent sets and scheduling, for which distributed algorithms are known in this model. Our algorithm produces an $O(\mu)$-approximate solution to the MST, and takes time $O(D \log n + \mu \log n)$, featuring a distributed scheduling complexity of $O(\mu \log n)$. Our main technical contribution is the use of *nearest neighbor trees*, which naturally ensure spatial separation at each step, thereby allowing SINR constraints to be satisfied.

Chapter 12

Conclusion and Discussion

In this dissertation, we study a set of fundamental problems in wireless networks under the physical interference model, with respect to throughput, delay and distributed complexity. An important basic factor in wireless networks is radio interference. The physical interference model, a widely accepted interference model which is considered more realistic than graph-based ones, has put a lot of challenges on designing efficient algorithms for distributed computation and cross-layer optimization. We study the connections between the two types of interference models, and provide the first set of solutions to the fundamental problems of low-complexity distributed scheduling and power control for dynamic traffic, maximum independent set, one-shot local broadcasting and minimum spanning tree, under the physical interference model. Besides, to optimize the system performance and to study delay-throughput tradeoffs, we take a "global" view across transport, network, MAC, physical layers and design a cross-layer optimization framework for delay-constrained throughput maximization under both of the graph-based and the physical interference models.

Our distributed scheduling and power control algorithms for dynamic traffic achieve a throughput region of $\Omega(\frac{1}{g(L)})\Lambda^{OPT}$ under SINR constraints. The algorithms are based on random access. We rigorously consider all aspects of the distributed complexity to properly analyze the performance.

In the rate-oblivious scheduling algorithm, our technique of using outdated infrequently-updated control info largely lowers the impact of local control information exchange to the efficiency ratio. However, this is at the cost of increased queueing delay, inferring a tradeoff between delay and achievable throughput region. Due to the complex form of the relation between delay and efficiency ratio, it is quite non-trivial yet useful to understand it and make decisions so that the right tradeoff emerge. We are still seeking efficient ways of extending the rate-oblivious scheduling (and power control) algorithm to multi-hop networks; a possible way is through the use of queue regulators [138]. In the current version, the information exchange step is synchronized, thus requiring a info-exchange sub-frame dedicated to exchanging control information in every frame. If we can synchronize the info-exchange step by embedding the control information in data packets or allow the control information to be updated in heterogeneous periods according to the queueing and traffic states, it is quite likely that we can increase the efficiency ratio. There are also rate-oblivious local scheduling schemes without information exchange among links, *e.g.*, adaptive CSMA mentioned in Section 3.1 which is proved to achieve an optimal efficiency ratio in theory; though there are some assumptions and it may not work as expected in every cases, it would be interesting to investigate if we can adapt some of those to the physical interference model with optimal efficiency ratio.

Our multi-commodity framework for delay-constrained throughput maximization for dynamic traffic result in approximation ratios independent of network size. It has many extensions and applications. For example, asynchronous random-access scheduling, multi-radio multi-channel networks, incorporating channel switching delays, cognitive networks and the physical interference model. Bounding end-to-end per-flow delay for maximal scheduling is an open and difficult problem. New progress on this will likely enable our DCTM framework to work with RA-SCHED-SINR, the maximal scheduling [106] and random-access maximal scheduling [49]. RB-SCHED-SINR suffers from high delays for low rate vectors, and our optimization framework cannot give good bounds if heterogeneous throughput rates are required. Combining both the random access and EDF-type scheduling schemes might be one way to address this problem.

Besides scheduling and power control for data transmission, the research in Chapter 9, 10 and 11 touches on another dimension of research in wireless networking, that of the complexity of distributed computation and the fundamental tradeoffs between performance and availability of technologies under the physical interference model. To our knowledge, our algorithms constitute the first set of fast distributed solutions in the setting of physical interference for the problems of maximum independent link set, maximum one-shot local broadcasting and minimum spanning tree. There remain a number of open problems, including power control, and improving and lower-bounding the running time. In Chapter 9, 10 and 11, we design distributed algorithms for the problems with a certain types of power assignments. If we are free to choose power assignments in a maximum independent set or a minimum spanning tree, optimum solutions under sub-linear length-monotone power assignments may be a logarithmic factor away from an optimum solution that considers power assignments as a dimension [53]. There exists a centralized algorithm that is order-optimal for the maximum independent set problem with power control [69]. However, it is not known how to make it in a distributed setting.

Distributed computing and cross-layer optimization in wireless networks under the physical interference model are still an emerging research area while being challenging due to the properties of SINR constraints. When we compare the graph-based interference models with the SINR model, it is known that solutions to the link scheduling problems developed under the graph-based model can be inefficient, if not infeasible, under the SINR model. For instance, Le et al. [85] show that the longest-queue-first scheme may result in zero throughput under SINR constraints (unlike that in the graph based model) for the case of dynamic traffic. As for MaxLSP, it is easy to show that when all the transmitters have uniform transmission/interference ranges, an optimum solution developed under a graph-based model may turn out to be a solution whose size is a fraction of $O\big((\frac{d_{max}}{d_{tx}})^2\big)$ of that of an optimum under the SINR model, where d_{max} is the length of the longest link and d_{tx} is the uniform transmission range. This is because that given a set of links under the SINR model, as long as all the senders are separated by cd_{max}, where c is some constant, all the links form an independent set; whereas in a feasible solution under the RBN model, senders need to be separated by a distance of $\Omega(d_{tx})$. Since we are dealing with an arbitrary topology, d_{max} may be small, leading to a much conservative solution under the SINR model.

We have extensively surveyed centralized solutions to various problems under the SINR model in this dissertation. Yet, it remains open to answer the following important fundamental questions in a quantitative way:

1. "What are the *gap* and *connection* between graph-based interference models and a physical interference model?"

2. "Is it *possible* to extend graph-based algorithms to the SINR setting? And *how*?"

More precisely, can we design a way to convert SINR-based interference constraints into graph-based ones, such that given a scheduling or routing algorithm that works under a graph-based model, we can adapt it to a physical interference model with throughput and delay performance guarantees? Understand the relationship between graph-based and physical interference models may help develop efficient solutions to some of the open problems under the physical interference model.

One step further, when it comes to distributed computing in the SINR model, we notice that the commonly studied model for distributed computing in wireless networks — the "Radio Broadcast Network (RBN)" model — which defines wireless node capabilities such as collision detection, is fundamentally different from the SINR-based distributed computing model. Efficient distributed algorithms are known in the RBN model for MaxLSP, as well as other fundamental problems such as coloring and dominating set, e.g., [4, 100, 118, 126]. Though it has not been rigorously proven, results from recent papers suggest this might not be feasible, or might only yield larger than constant factor gaps. For instance, Chafekar et al. [24] discuss an instance where the solution in the "equivalent" RBN model could be significantly smaller than that in the SINR model; see also [99]. In other words, a distributed algorithm in the RBN model cannot be implemented in general under the SINR-based model. Further, the distributed wireless communication mechanism can be quite different. For example, the RBN model does not allow for capabilities to determine the signal strength and make decisions based on that. Given the fundamental differences between the models, more formal analysis of the SINR model, *e.g.*, the SINR diagrams [13,14] and the space-time SINR random graph [18], is needed.

In Chapters 9 and 10, we have covered the design of distributed algorithms *w.r.t.* different levels of radio technology availability, including full-/half-duplex, adaptive/non-adaptive power radios. We showed that the employment of more advanced communication technologies have enabled us to design more efficient algorithms in distributed computation. It would also be interesting if we incorporate more advanced technologies dealing with wireless interference — for example, using multi-channel multi-radio (MCMR) — in the design of distributed algorithms and quantify the performance gains.

Bibliography

[1] E. Aarts, R. Harwig, and M. Schuurmans. Ambient intelligence. In P. Denning, editor, *The Invisible Future: The Seamless Integration Of Technology Into Everyday Life*, pages 235–250. McGraw-Hill Companies, 2001.

[2] T. Achterberg. Scip: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[3] Y. Afek, N. Alon, Z. Bar-Joseph, A. Cornejo, B. Haeupler, and F. Kuhn. Beeping a maximal independent set. In *DISC*, 2011.

[4] Y. Afek, N. Alon, O. Barad, E. Hornstein, N. Barkai, and Z. Bar-Joseph. A biological solution to a fundamental distributed computing problem. *Science*, 331:183–185, 2011.

[5] N. Ahmed, U. Ismail, S. Keshav, and K. Papagiannaki. Online estimation of rf interference. In *ACM CoNEXT*, pages 4:1–4:12, 2008.

[6] R. Ahuja, R. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[7] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50:2127–2159, 2006.

[8] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *Journal of Computer and System Sciences*, 43:290–298, 1991.

[9] C. Ambuhl. An optimal bound for the MST algorithms to compute energy efficient broadcast trees in wireless networks. In *Proceedings of the 32th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1139–1150, November 2005.

[10] M. Andrews and L. Zhang. Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule. In *IEEE INFOCOM*, 1999.

[11] M. Andrews and L. Zhang. Satisfying arbitrary delay requirements in multihop networks. In *IEEE INFOCOM*, 2008.

[12] E. I. Ásgeirsson and P. Mitra. On a game theoretic approach to capacity maximization in wireless networks. In *IEEE INFOCOM*, 2011.

[13] C. Avin, Y. Emek, E. Kantor, Z. Lotker, D. Peleg, and L. Roditty. Sinr diagrams: towards algorithmically usable sinr models of wireless networks. In *ACM symposium on Principles of distributed computing (PODC)*, pages 200–209, 2009.

[14] C. Avin, Y. Emek, E. Kantor, Z. Lotker, D. Peleg, and L. Roditty. Sinr diagrams: Convexity and its applications in wireless networks. *Journal of ACM*, 59(4):18:1–18:34, 2012.

[15] C. Avin, Z. Lotker, F. Pasquale, and Y. A. Pignolet. A note on uniform power connectivity in the sinr model. In *ALGOSENSORS*, pages 116–127, 2009.

[16] B. Awerbuch, A. Bar-Noy, and M. Gopal. Approximate distributed bellman-ford algorithms. *IEEE Transactions on Communications*, 42(8):2515–2517, 1994.

[17] B. Awerbuch, M. Luby, A. V. Goldberg, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 364–369, 1989.

[18] F. Baccelli, B. Blaszczyszyn, and M. O. H. Mirsadeghi. Optimal paths on the space-time sinr random graph. *Advances in Applied Probability*, 43(1):131–150, 2011.

[19] G. Baier, T. Erlebach, A. Hall, E. Köhler, H. Schilling, and M. Skutella. Length-bounded cuts and flows. In *ICALP (1)*, pages 679–690, 2006.

[20] H. Balakrishnan, C. Barrett, A. Kumar, M. Marathe, and S. Thite. The Distance 2-Matching Problem and Its Relationship to the MAC Layer Capacity of Adhoc Wireless Networks. *special issue of IEEE J. Selected Areas in Communications*, 22(6):1069–1079, 2004.

[21] A. S. Bedekar and M. Azizoglu. On the information-theoretic capacity of discrete-time queues. *IEEE Trans. on Info. Theory*, 44:446–461, 1998.

[22] N. Bisnik and A. Abouzeid. Queuing network models for delay analysis of multihop wireless ad hoc networks. *Ad Hoc Networks*, 7(1), 2009.

[23] L. X. Bui, S. Sanghavi, and R. Srikant. Distributed link scheduling with constant overhead. *IEEE/ACM Trans. on Netw.*, 17(5):1467–1480, 2009.

[24] D. Chafekar, V. S. A. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Approximation algorithms for computing capacity of wireless networks with sinr constraints. In *INFOCOM 2008*, pages 1166 –1174, 2008.

[25] D. Chafekar, V. S. A. Kumar, M. V. Marathe, S. P. 0002, and A. Srinivasan. Cross-layer latency minimization in wireless networks with sinr constraints. In *ACM MobiHoc*, pages 110–119, 2007.

[26] D. Chafekar, D. Levin, V. S. A. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Capacity of asynchronous random-access scheduling in wireless networks. In *IEEE INFOCOM*, 2008.

[27] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar. Throughput and fairness guarantees through maximal scheduling in wireless networks. *IEEE Trans. on Info. Theory*, 54 (2):572–594, 2008.

[28] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar. Throughput and fairness guarantees through maximal scheduling in wireless networks. *IEEE Trans. on Info. Theory*, 54 (2):572–594, 2008.

[29] B. Clark, C. Colbourn, and D. Johnson. Unit disk graphs. *Discrete Mathematics*, 86: 165–177, 1990.

[30] A. Clementi, P. Crescenzi, P. Penna, G. Rossi, and P. Vocca. On the complexity of computing minimum energy consumption broadcast subgraph. In *Proceedings of the*

18th Annual Symposium on Theoretical Aspects of Computer Science (STACS), pages 121–131, June 2001.

[31] R. Cole and U. Vishkin. Deterministic coin tossing and accelerating cascades: micro and macro techniques for designing parallel algorithms. In *ACM symposium on Theory of computing (STOC)*, pages 206–219, 1986.

[32] M. Dinitz. Distributed algorithms for approximating wireless network capacity. In *IEEE INFOCOM*, 2010.

[33] M. Draief, J. Mairesse, and N. O'Connell. Queues, stores, and tableaux. *Journal of Applied Probability*, 42(4):1145–1167, 2005.

[34] M. Durvy, O. Dousse, and P. Thiran. Border effects, fairness, and phase transition in large wireless networks. In *IEEE INFOCOM*, pages 601–609, 2008.

[35] M. Elkin. An overview of distributed approximation. *ACM SIGACT News Distributed Computing Column*, 35(4):40–57, December 2004.

[36] A. Fanghänel, S. Geulen, M. Hoefer, and B. Vöcking. Online capacity maximization in wireless networks. In *Proceedings of the 22nd ACM symposium on Parallelism in algorithms and architectures*, SPAA '10, pages 92–99, 2010.

[37] A. Fanghänel, T. Kesselheim, and B. Vöcking. Improved algorithms for latency minimization in wireless networks. In *ICALP '09: Proceedings of the 36th Internatilonal Collogquium on Automata, Languages and Programming*, pages 447–458, 2009.

[38] R. Gallager, P. Humblet, and P. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems*, 5(1): 66–77, January 1983.

[39] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. In *IEEE INFOCOM*, 2004.

[40] J. Garay, S. Kutten, and D. Peleg. A sublinear time distributed algorithm for minimum-weight spanning trees. *SIAM Journal of Computing*, 27:302–316, 1998.

[41] L. Gasieniec, D. Peleg, and Q. Xin. Faster communication in known topology radio networks. In *Proceedings of the 24th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 129–137, 2005.

[42] L. Georgiadis, R. Guerin, V. Peris, , and K. Sivarajan. Efficient network qos provisioning based on per node traffic shaping. In *IEEE INFOCOM*, 1996.

[43] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Netw.*, 1(1):1–149, 2006.

[44] O. Goussevskaia, M. Halldórsson, R. Wattenhofer, and E. Welzl. Capacity of arbitrary wireless networks. In *IEEE INFOCOM*, 2009.

[45] O. Goussevskaia, T. Moscibroda, and R. Wattenhofer. Local Broadcasting in the Physical Interference Model. In *ACM SIGACT-SIGOPT International Workshop on Foundations of Mobile Computing (DialM-POMC)*, 2008.

[46] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric sinr. In *ACM MOBIHOC*, 2007.

[47] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric sinr. In *Proceedings of MOBIHOC*, pages 100–109, 2007.

[48] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. on Netw.*, 10:477–486, 2002.

[49] A. Gupta, X. Lin, and R. Srikant. Low-complexity distributed scheduling algorithms for wireless networks. *IEEE/ACM Trans. on Netw.*, 17(6):1846–1859, 2009.

[50] G. R. Gupta and N. B. Shroff. Delay analysis for multi-hop wireless networks. In *IEEE INFOCOM*, 2009.

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

[52] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 19–28, 1999.

[53] Halldórsson and Mitra. Wireless capacity with oblivious power in general metrics. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[54] M. M. Halldórsson and P. Mitra. Nearly optimal bounds for distributed wireless scheduling in the sinr model. In *ICALP*, 2011.

[55] M. M. Halldórsson and R. Wattenhofer. Wireless communication is in apx. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2009.

[56] B. Han, V. S. A. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Distributed strategies for channel allocation and scheduling in software-defined radio networks. In *IEEE INFOCOM*, pages 1521–1529, 2009.

[57] J. Hsu and P. Burke. Behavior of tandem buffers with geometric input and markovian output. *IEEE Trans. on Comm.*, 24(3):358–361, 1976.

[58] K.-L. Hung and B. Bensaou. Distributed rate control and contention resolution in multi-cell ieee 802.11 wlans with hidden terminals. In *ACM Mobihoc*, pages 51–60, 2010.

[59] S. Jagabathula and D. Shah. Optimal delay scheduling in networks with arbitrary constraints. In *ACM SIGMETRICS*, 2008.

[60] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha. Practical, real-time, full duplex wireless. In *ACM MobiCom*, pages 301–312, 2011.

[61] J. J. Jaramillo, R. Srikant, and L. Ying. Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints. *IEEE Journal on Selected Areas in Communications*, 29(5), 2011.

[62] P. Jayachandran and M. Andrews. Minimizing end-to-end delay in wireless networks using a coordinated edf schedule. In *IEEE Infocom*, 2010.

[63] L. Jiang and J. Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. *IEEE/ACM Trans. Netw.*, 18:960–972, 2010.

[64] C. Joo, X. Lin, and N. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithm in multihop wireless networks. *IEEE/ACM Trans. on Netw.*, 17 (4):1132–1145, 2009.

[65] C. Joo and N. B. Shroff. Performance of random access scheduling schemes in multi-hop wireless networks. *IEEE/ACM Trans. on Netw.*, 17(5):1481–1493, 2009.

[66] K. Kar, X. Luo, and S. Sarkar. Asymptotically tight delay guarantees through wireless link scheduling. In *IEEE INFOCOM*, 2009.

[67] K. Kar, S. Sarkar, A. Ghavami, and X. Luo. Delay guarantees for throughput-optimal wireless link scheduling. *IEEE Transactions on Automatic Control*, 57(11):2906–2911, Nov. 2012.

[68] I. Keslassy and N. McKeown. Analysis of scheduling algorithms that provide 100% throughput in input-queued switches. In *Allerton Conference on Communication, Control, and Computing*, 2001.

[69] T. Kesselheim. A constant-factor approximation for wireless capacity maximization with power control in the sinr model. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011.

[70] T. Kesselheim and B. Vöcking. Distributed contention resolution in wireless networks. In *International Symposium on Distributed Computing (DISC)*, 2010.

[71] M. Khan and G. Pandurangan. A fast distributed approximation algorithm for minimum spanning trees. *Distributed Computing*, 20(6):391–402, Apr. 2008. **Invited paper**, Best Student Paper Award in DISC 2006.

[72] M. Khan, G. Pandurangan, and V. Kumar. A simple randomized scheme for constructing low-weight k-connected spanning subgraphs with applications to distributed algorithms. *Theoretical Computer Science*, 385(1-3):101–114, Oct. 2007.

[73] M. Khan, G. Pandurangan, and V. Kumar. Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(1):124–139, 2009.

[74] M. Khan, G. Pandurangan, G. Pei, and A. K. S. Vullikanti. Brief announcement: A fast distributed approximation algorithm for minimum spanning trees in the sinr model. In *International Symposium on DIStributed Computing (DISC)*, 2012.

[75] M. Khan, G. Pandurangan, G. Pei, and A. K. S. Vullikanti. A fast distributed approximation algorithm for minimum spanning trees in the sinr model, 2012. http://arxiv.org/abs/1206.1113.

[76] M. A. Khojastepour, K. Sundaresan, S. Rangarajan, X. Zhang, and S. Barghi. The case for antenna cancellation for scalable full-duplex wireless communications. In *ACM Workshop on Hot Topics in Networks (HotNets)*, pages 17:1–17:6, 2011.

[77] M. Kodialam and T. Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *ACM MobiCom*, pages 42–54, 2003.

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

[79] S. Kompella, S. Mao, Y. Hou, and H. Sherali. On path selection and rate allocation for concurrent video sessions in ad hoc networks. *IEEE/ACM Trans. on Netw.*, 17(1): 212–224, 2009.

[80] V. S. A. Kumar, M. V. Marathe, and S. Parthasarathy. Cross-layer capacity estimation and throughput maximization in wireless networks. In G. Cormode and M. Thottan, editors, *Algorithms for Next Generation Networks*, Computer Communications and Networks, pages 67–98. Springer London, 2010.

[81] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end packet-scheduling in wireless ad-hoc networks. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1021–1030, 2004.

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

[83] P. Kyasanur and N. H. Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *MobiCom*, pages 43–57, 2005.

[84] P. Kyasanur and N. H. Vaidya. Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 10(1):31–43, January 2006.

[85] L. Le, E. Modiano, C. Joo, and N. B. Shroff. Longest-queue-first scheduling under sinr interface model. In *ACM MOBIHOC*, 2010.

[86] L. B. Le, K. Jagannathan, and E. Modiano. Delay analysis of maximum weight scheduling in wireless ad hoc networks. In *Annual Conference on Information Sciences and Systems (CISS)*, pages 389–394, 2009.

[87] L. B. Le and R. R. Mazumdar. Control of wireless networks with flow level dynamics under constant time scheduling. *Wirel. Netw.*, 16:1355–1372, 2010.

[88] H.-W. Lee, E. Modiano, and L. B. Le. Distributed throughput maximization in wireless networks via random power allocation. In *WiOPT'09: Proceedings of the 7th international conference on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 328–336, 2009.

[89] C. Li and E. Knightly. Coordinated network scheduling: A framework for end-to-end services. In *IEEE ICNP*, 2000.

[90] C. Li and E. Knightly. Schedulability criterion and performance analysis of coordinated schedulers. In *ITC*, 2001.

[91] X. Lin and S. Rasool. A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad-hoc wireless networks. In *IEEE INFOCOM*, pages 1118–1126, 2007.

[92] X. Lin and S. Rasool. Constant-time distributed scheduling policies for ad hoc wireless networks. *IEEE Transactions on Automatic Control*, 54(2):231–242, 2009.

[93] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Trans. on Netw.*, 14(2):302–315, 2006.

[94] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor. Towards utility-optimal random access without message passing. *Wirel. Commun. Mob. Comput.*, 10:115–128, 2010.

[95] Z. Lotker and D. Peleg. Structure and algorithms in the sinr wireless model. *ACM SIGACT News*, 41(2), 2010.

[96] M. Luby. A simple parallel algorithm for the maximal independent set problem. In *ACM STOC*, 1985.

[97] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, 1996.

[98] Y. Metivier, J. M. Robson, N. Saheb-Djahromi, and A. Zemmari. An optimal bit complexity randomized distributed mis algorithm. In *SIROCCO*, pages 323–337, 2009.

[99] T. Moscibroda. The worst-case capacity of wireless sensor networks. In *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–10, 2007.

[100] T. Moscibroda and R. Wattenhofer. Maximal independent sets in radio networks. In *ACM symposium on Principles of distributed computing (PODC)*, 2005.

[101] T. Moscibroda and R. Wattenhofer. The complexity of connectivity in wireless networks. In *IEEE INFOCOM*, 2006.

[102] T. Moscibroda and R. Wattenhofer. The Complexity of Connectivity in Wireless Networks. In *25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Barcelona, Spain*, April 2006.

[103] T. Moscibroda, R. Wattenhofer, and A. Zollinger. Topology control meets sinr: the scheduling complexity of arbitrary topologies. In *ACM Mobihoc*, pages 310–321, 2006.

[104] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[105] B. Nardelli, J. Lee, K. Lee, Y. Yi, S. Chong, E. Knightly, and M. Chiang. Experimental evaluation of optimal csma. In *IEEE INFOCOM*, 2011.

[106] M. J. Neely. Delay analysis for maximal scheduling with flow control in wireless networks with bursty traffic. *IEEE/ACM Trans. on Netw.*, 17(4):1146–1159, 2009.

[107] M. J. Neely and E. Modiano. Capacity and delay tradeoffs for ad-hoc mobile networks. *IEEE Trans. on Information Theory*, 51:1917–1937, 2005.

[108] J. Ni, B. Tan, and R. Srikant. Q-csma: queue-length based csma/ca algorithms for achieving maximum throughput and low delay in wireless networks. In *IEEE INFOCOM (Mini)*, pages 271–275, 2010.

[109] A. Panconesi and A. Srinivasan. Improved distributed algorithms for coloring and network decomposition problems. In *ACM symposium on Theory of computing (STOC)*, pages 581–592, 1992.

[110] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Trans. on Netw.*, 2(2), 1994.

[111] G. Pei and V. A. Kumar. Low-complexity scheduling for wireless networks. In *ACM Mobihoc*, 2012.

[112] G. Pei, V. S. A. Kumar, S. Parthasarathy, and A. Srinivasan. Approximation algorithms for throughput maximization in wireless networks with delay constraints. In *INFOCOM*, pages 1116–1124, 2011.

[113] G. Pei and A. K. S. Vullikanti. Brief announcement: Distributed algorithms for maximum link scheduling under the physical interference model. In *International Symposium on DIStributed Computing (DISC)*, 2012.

[114] G. Pei and A. K. S. Vullikanti. Distributed algorithms for maximum link scheduling in the physical interference model, 2012. http://filebox.vt.edu/users/somehi/papers/maxlsp12.pdf.

[115] G. Pei and A. K. S. Vullikanti. Distributed algorithms for maximum link scheduling under the physical interference model, 2012. http://arxiv.org/abs/1208.0811.

[116] G. Pei and A. K. S. Vullikanti. Distributed approximation algorithms for link scheduling and local broadcasting in the physical interference model, 2012. Technical Report, Virginia Bioinformatics Institute: http://filebox.vt.edu/users/somehi/papers/maxlspnonuniform12.pdf.

[117] G. Pei and A. K. S. Vullikanti. Throughput maximization for secondary-users with delay guarantees in cognitive networks under the physical interference model, 2012. Technical Report.

[118] D. Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, 2000.

[119] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2000.

[120] A. Proutiere, Y. Yi, T. Lan, and M. Chiang. Resource allocation over network dynamics without timescale separation. In *IEEE INFOCOM*, pages 406–410, 2010.

[121] S. Rajagopalan and D. Shah. Distributed algorithm and reversible network. In *CISS*, pages 498–502, 2008.

[122] S. Ramanathan and E. L. Lloyd. Scheduling algorithms for multihop radio networks. *IEEE/ACM Trans. on Netw.*, 1:166–177, 1993.

[123] S. Ray, D. Starobinski, and J. Carruthers. Performance of wireless networks with hidden nodes: A queueing-theoretic analysis. *Journal of Computer Communications*, 28:1179–1192, 2005.

[124] A. Reddy, S. Banerjee, A. Gopalan, S. Shakkottai, and L. Ying. Wireless scheduling with heterogeneously delayed network-state information. In *Allerton Conference*, pages 1577–1584, 2010.

[125] C. Scheideler, A. Richa, and P. Santi. An $o(logn)$ dominating set protocol for wireless ad-hoc networks under the physical interference model. In *ACM MobiHoc*, pages 91–100, 2008.

[126] J. Schneider and R. Wattenhofer. A log* distributed maximal independent set algorithm for growth-bounded graphs. In *ACM symposium on Principles of distributed computing (PODC)*, 2005.

[127] G. Sharma and R. Mazumdar. Scaling laws for capacity and delay in wireless networks with random mobility. In *IEEE ICC*, 2004.

[128] G. Sharma, R. Mazumdar, and N. B.Shroff. Delay and capacity trade-offs in mobile ad hoc networks: a global perspective. *IEEE/ACM Trans. on Netw.*, 15(5):981–992, 2007.

[129] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra. Centaur: realizing the full potential of centralized wlans through a hybrid data path. In *Mobicom*, pages 297–308, 2009.

[130] S. Srinivasa and M. Haenggi. Taseps: A statistical mechanics tool to analyze the performance of wireless line networks. submitted to IEEE/ACM Trans. on Netw., 2009.

[131] A. Srinivasan. An extension of the Lova´sz Local Lemma, and its applications to integer programming. *SIAM J. Comput.*, 36(3):609–634, September 2006.

[132] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks. *IEEE Trans. on Automatic Control*, 37(12):1936–1949, 1992.

[133] S. Toumpis and A. Goldsmith. Large wireless networks under fading, mobility, and delay constraints. In *IEEE INFOCOM*, pages 609–619, 2004.

[134] P. Wan, G. Calinescu, X. Li, and O. Frieder. Minimum-energy broadcasting in static ad hoc wireless networks. *Wireless Networks*, 8(6):607–617, November 2002.

[135] P.-J. Wan. Multiflows in multihop wireless networks. In *MobiHoc*, pages 85–94, 2009.

[136] P.-J. Wan, O. Frieder, X. Jia, F. F. Yao, X. Xu, and S. Tang. Wireless link scheduling under physical interference model. In *IEEE INFOCOM*, pages 838–845, 2011.

[137] P.-J. Wan, X. Jia, and F. Yao. Maximum independent set of links under physical interference model. In *Proceedings of WASA*, pages 169–178, 2009.

[138] X. Wu, R. Srikant, and J. R. Perkins. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. In *IEEE INFOCOM*, 2006.

[139] R. Wunderling. *Paralleler und objektorientierter Simplex-Algorithmus*. PhD thesis, Technische Universität Berlin, 1996.

[140] A. Wyglinski, M. Nekovee, and Y. Hou, editors. *Cognitive Radio Communications and Networks: Principles and Practice*. Academic Press/Elsevier, 2010.

[141] M. Xie and M. Haenggi. Towards an end-to-end delay analysis of wireless multihop networks. *Ad Hoc Networks*, 7(5):849–861, 2009.

[142] L. Ying and S. Shakkottai. On throughput optimality with delayed network-state information. *IEEE Trans. on Info. Theory*, 57:5116–5132, 2011.

[143] C. Zhao and X. Lin. On the queue-overflow probabilities of distributed scheduling algorithms. In *IEEE CDC*, pages 4820–4825, 2009.