Capacity of Wireless Networks under SINR Interference Constraints

Deepti Chafekar * V. S. Anil Kumar † Madhav V. Marathe ‡ Srinivasan Parthasarathy§
Aravind Srinivasan ¶

June 21, 2011

Abstract

A fundamental problem in wireless networks is to estimate their throughput capacity - given a set of wireless nodes and a set of connections, what is the maximum rate at which data can be sent on these connections. Most of the research in this direction has focused either on random distributions of points, or has assumed simple graph-based models for wireless interference. In this paper, we study the capacity estimation problem using a realistic Signal to Interference Plus Noise Ratio (SINR) model for interference, on arbitrary wireless networks without any assumptions on node distributions. The problem becomes much more challenging for this setting, because of the non-locality of the SINR model. Recent work by Moscibroda et al. (IEEE INFOCOM 2006, ACM MobiHoc 2006) has shown that the throughput achieved by using SINR models can differ significantly from that obtained by using graph-based models. In this work, we develop *polynomial time* algorithms to provably approximate the throughput capacity of wireless network under the SINR model.

1 Introduction

A fundamental problem in wireless networks is to estimate their throughput capacity - given a set of wireless nodes, and a set of end-to-end (multi-hop) connections, what is the maximum rate at which data can be sent on these connections? Starting with [11], there has been a large body of work on this problem, especially for networks formed by a random distribution of nodes in the unit square. A related and more practical question is to estimate the capacity of the given network, and develop protocols to utilize the network close to its capacity. This question becomes difficult in wireless networks because of interference, which constrains the set of links that can transmit simultaneously. The algorithmic aspects of network capacity have been studied in a number of papers, such as [2, 13–15, 17, 26].

A commonly used approach when designing *provably good algorithms* is to represent the underlying wireless network as a geometric intersection graph. For a set of nodes V on an Euclidean plane, each node $u \in V$ is associated with a disk of radius $range(u)$, which depends on the transmission power level $J(u)$ of node u; a common approximation is to choose $range(u) = \Theta((J(u))^{1/\alpha})$, where α is the path-loss exponent, and the signal from node u is assumed to be heard only within this range. This gives us the *connectivity graph* $G = (V, E)$ where set E is obtained by adding links from u to v, for all $u, v \in V$, if $d(u, v) \leq range(u)$, where $d(u, v)$ denotes the Euclidean distance from node u to node v. Interference in such a graph is modeled through *independence constraints* (see e.g., [25]): *if a node u transmits, no node in its vicinity can transmit*. A number of papers have studied MAC protocols with these geometric models of interference [25, 26]. Intuitively, such graph-based models make the algorithmic analysis tractable since they localize the interference effect of a transceiver on others.

*Nokia Research, chafekar@nokia.com; part of this work done while at Virginia Tech; research supported in part by NSF Award CNS-0626964.
†Virginia Tech., akumar@vbi.vt.edu; supported in part by NSF Award CNS-0626964.
‡Virginia Tech., mmarathe@vbi.vt.edu; supported in part by NSF Award CNS-0626964.
§IBM Research, spartha@us.ibm.com
¶University of Maryland, College Park, srin@cs.umd.edu; A. Srinivasan was supported in part by NSF ITR Award CNS-0426683, NSF Award CNS-0626964, NSF Award CNS-0626636, and NSF Award CNS 1010789; part of this work was done while he was on sabbatical at the Network Dynamics and Simulation Science Laboratory of the Virginia Bioinformatics Institute, Virginia Tech.
[0]A preliminary version of this paper appeared in the Proc. of INFOCOM 2008 [8].

While such graph-based models give a useful first approximation to understanding wireless networks, they have several limitations. Graph-based models assume that the signal from a radio can only be heard within its range and that signal collisions always lead to lost messages. These assumptions oversimplify the real process. A more realistic model that has been used to study wireless transmission is called the *Signal to Interference Plus Noise Ratio (SINR)* model [11, 23]: a signal from a transmitter u is successfully received by a receiver at v, if the *ratio* of u's signal strength at v and the combined interference from other transmitters along with ambient noise exceeds v's antenna gain. In other words, a set of transmissions $e_j = (u_j, v_j), \forall j = \{1, \ldots, k\}$, where node u_j transmits to node v_j can be simultaneously scheduled if for all $e_i = (u_i, v_i)$ we have,

$$SINR(v_i) = \frac{J(e_i)}{d(u_i, v_i)^\alpha \left[N_0 + \sum_{j \neq i} \frac{J(e_j)}{d(u_j, v_i)^\alpha}\right]} \geq \beta, \tag{1}$$

where N_0 denotes the noise density, α denotes the path-loss exponent, β is the antenna gain, $d(u_i, v_i)$ denotes the Euclidean distance between nodes u_i, v_i and $J(e_i)$ denotes the power level with which node u_i transmits. Recent work by Moscibroda et al. [21–23] has shown that for several problems, this model is significantly different from graph-based models. In [21, 23], they show that for the problem of minimizing the *scheduling complexity*, by choosing appropriate transmission power levels, SINR models can yield much shorter schedules. In [22], they show that the throughput capacity under a SINR model is different from that under a graph-based model. The non-locality of this model makes its analysis challenging.

In this paper we consider the problem of characterizing the achievable rates for arbitrary multi-hop wireless networks with SINR constraints. ***The central contribution of this work is the solution to the following cross-layer optimization problem: given a set of wireless transmitters, set of connections and a power assignment for all transmission links in the network, what should be the rate at which data is sent from every source, how are the packest routed through the network, and how should the transmissions across the links be scheduled, so that the network-wide throughput (or a throughput related objective) is optimized?*** The formal description of the problem is as follows: Given a set of nodes V, a set of possible directed edges E, a set of source-destination pairs $\mathcal{D} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$, and a power level vector $\bar{J}$ that specifies power level $J(e)$ for transmission on edge e, the throughput maximization problem with SINR constraints (TM-SINR) consists of

- choosing routes for the connections (Routing),
- choosing flow rates on the routes (Rate Control), and
- choosing which links to schedule at each time slot, so that the SINR constraints are satisfied for all simultaneous transmissions (Scheduling).

We note that the TM-SINR problem does not involve power control, i.e., the power levels $J(e)$ for each edge are fixed and given as part of the input (e.g., the power levels could be determined based on energy consumption requirements or other network life-time requirements by a higher-layer application).

The SINR constraints make the throughput optimization problem *non-convex*. Further the *link scheduling* problem with SINR constraints has been shown to be NP-hard in [10]. Since scheduling is also an integral component of our problem, it is reasonable to conjecture that the TM-SINR problem is also NP-hard. We therefore focus on developing rigorous *polynomial time* approximation algorithms for our problem with *provable performance guarantees*.

In reality, the link capacities depend on the SINR [1], thereby making this problem very complex. We simplify this by using the Additive White Gaussian Noise (AWGN) model for specifying the link capacities [5]. In this model the capacity $cap(e)$ of a link $e = (u, v)$ having length $\ell(e) = d(u, v)$, where $d(u, v)$ is the Euclidean distance between nodes u, v, and transmitting at power level $J(e)$ is given by

$$cap(e) = W \log_2 \left(1 + \frac{J(e)}{\ell(e)^\alpha N_0 W}\right), \tag{2}$$

where W is the bandwidth, and N_0 and α are as defined following equation 1. In the absence of interference, the above equation provides a theoretical upper bound on the link capacity. The maximum throughput problem with SINR constraints is significantly challenging even under the AWGN model and we focus on this model in this work.

2 Overview of results

We rigorously study the TM-SINR problem in wireless networks, and take the first steps toward developing efficient algorithms for this problem. The main contributions of our work are summarized below.

1. We compare the SINR and graph-based models for the same instance, with the same fixed power levels, and observe that the throughput capacity can be significantly different in these two models. When the power level for all the edges is the same, we show that there are instances in which the throughput capacity that can be achieved in the SINR model is significantly higher than that achieved in the graph-based model. For the case of *linear* power levels (where $J(e) \propto \ell(e)^\alpha$ for each edge e), we show that there are instances in which the throughput capacity achieved in the SINR model can be much lower than that achieved in the corresponding graph-based model with the same power levels. In contrast, the results of [21–23] show that by choosing suitable power levels, a much higher throughput capacity is possible in SINR models than in graph-based models. Since all these models of interference are approximations of the reality, this suggests greater care is needed in inferring any properties of the system based on such an analysis.

2. We develop a linear programming based approach to approximate the maximum throughput rate vector in the case of SINR constraints. For the generic case of non-uniform power levels, in which the power levels on different edges could be different, we develop a *polynomial time* approximation algorithm that provides a feasible rate vector whose total throughput is at least $O(r_{opt}/\log\Delta\cdot\log\Gamma^1)$. Here r_{opt} is the maximum possible throughput for any particular instance of the TM-SINR problem, Δ is defined as $\Delta = \max_{e\in E}\ell(e)/\min_{e'\in E}\ell(e')$ where $\ell(e)$ denotes the length of the edge e, and Γ is the ratio between maximum and minimum power levels used. This gives us an $O(\log\Delta\cdot\log(\Gamma))$ approximation to the total throughput. Our approximation bound is a worst case guarantee that holds for every instance. To our knowledge, this is the first such provably good polynomial-time capacity estimation and throughput maximization algorithm for arbitrary networks under SINR interference constraints.

3. We consider two special cases of power level choices - *uniform* power level choice (when all nodes have the same power level J) and *linear* power level choice (where the power level on each edge $e = (u, v)$ such that $\ell(e) = d(u, v)$ is $J(e) = c_1 \cdot \ell(e)^\alpha$ for a constant c_1). For both the cases we improve the throughput approximation to $O(\log\Delta)$.

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

5. Finally, we show that a constant ($O(1)$) factor approximation can be achieved to the maximum throughput for the special case where the topology is a grid.

A preliminary version of this paper appeared in [8], and covered the first three results mentioned above. In this paper, we additionally conduct extensive simulations that validate and improve upon the theoretical results in practical instances. We use both random and realistic topologies to understand the approximation bounds of our algorithm. We view the TM-SINR problem as a **bi-criteria** approximation problem, wherein by adding certain constraints to the input power level, we ensure that our solution is poly-log factor away from the optimal solution. For a given instance of the TM-SINR problem, we obtain a solution that uses a slightly higher power level on every transmitting edge than the power level used by the respective optimum solution. By increasing the power levels by a factor of $1+\epsilon$, where ϵ is a small positive slack, we ensure that the rate vectors obtained by our solution can be scheduled feasibly. This is explained formally in Section 6.3. Our algorithm builds upon the recent work of [7, 21, 23] on scheduling with SINR constraints, and the LP based approaches of [14, 17] for estimating the capacity for graph-based interference models. We note that the primary focus of this work is theoretical, and our framework does not incorporate all the aspects of different protocols. However, it can help in obtaining absolute performance limits on the system, which can help in evaluating real protocols.

[1] All the logarithms considered in this paper are to the base two and floor of the log functions is considered, for e.g. $\log\Delta$ is actually considered as $\lfloor\log\Delta\rfloor$.

3 Related Work

There has been significant work on understanding the capacity of random networks using both graph-based and SINR models (see, e.g., [4, 11, 16]). However, these results do not directly help in understanding the capacity of arbitrary networks, which is the focus of our paper. The throughput maximization problem for graph-based models is formally studied and proven to be NP-hard by Jain *et. al.* [13], who use a linear programming approach to characterize the capacity of the network and to perform routing. They model interference constraints as a *conflict graph* and provide upper and lower bounds for optimal throughput. As mentioned in [6], the methods discussed in [13] tend to have an exponential complexity and no *performance guaranteed polynomial time* approximation algorithm is proposed. Toumpis *et. al.* [28] provide a mathematical framework for determining the capacity region of an ad-hoc network, which captures the effects of power control, spatial reuse and successive interference cancellation on the capacity region. However, their results do not give worst case approximation guarantees.

Kodialam *et. al.* [14] study the problem of determining achievable rates for multi-hop wireless networks, along with joint routing and scheduling constraints in graph-based models. Their approach provides necessary and sufficient conditions for link flows and leads to a *polynomial time* approximation algorithm for this problem. However, they only consider *primary* interference in their model, which is very restrictive. Lin et. al. [19, 20] study the joint problem of rate allocation and scheduling using a dual optimization based approach to decompose the problem as rate control and scheduling problem. Their technique provides an optimal solution that maximizes the throughput and provides a stable and fair schedule considering the primary interference model. Although some of these approximation bounds have been improved in recent work by Buragohain *et. al.* [6], it is not clear how these techniques could be extended to the SINR interference model.

Some of the key algorithmic results on the SINR model are studied in [7, 21–23]. Moscibroda et al. [21–23] study the problem of scheduling edges with SINR constraints to ensure that some property (e.g., connectivity) is satisfied by the edges that are chosen. They show that by suitable power control, the solutions in the SINR model are much more efficient than those in graph-based models. Chafekar et al. [7] develop approximation algorithms for packet scheduling to minimize end-to-end delays with SINR constraints.

Our work is closely related to [7, 9, 17]. The work by [17] provides a constant approximation algorithm for the throughput maximization problem along with joint scheduling and routing. The interference model considered is graph-based and their approach is generic enough to accommodate the case of uniform and non-uniform power levels. They further derive linear necessary and sufficient conditions that lead to a constant factor approximation to the throughput capacity. However, the framework presented in [17] cannot be easily extended to the SINR interference model. Chafekar et al. [7] design a polylogarithmic approximation algorithm for the problem of end-to-end latency minimization in the SINR model, which was improved by Fanghanel et al. [9]. In this work, by combining and extending some of the techniques from [7, 17], we study the throughput maximization problem along with joint routing and scheduling for the SINR interference constraints.

4 Preliminaries

We consider the input instance of the TM-SINR problem to be specified as $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$, where (i) V denotes a set of transceivers henceforth referred to as nodes, which are located on a Euclidean plane, (ii) $E \subseteq V \times V$ denote the set of possible directed links (also referred to as edges), on which transmissions can occur, (iii) $\mathcal{D}$ is a set of connections, with the ith connection from node s_i to node t_i, and (iv) $\bar{J} = (J(e) : e = (u, v) \in E)$ specifies the vector of power transmission levels on edges. For $u, v \in V$, let $d(u, v)$ denote the Euclidean distance between these nodes; for $e = (u, v) \in E$, let $\ell(e) = d(u, v)$. Following standard graph theory notation, let $N_{out}(u)$ and $N_{in}(u)$ be the sets of outgoing and incoming edges for node u respectively. Let

$$\Delta = \max_{e \in V}\{\ell(e)\}/\min_{e' \in E}\{\ell(e')\};$$

where $\log \Delta$ is also called the "length diversity" [10]. Also, let

$$\Gamma = \frac{\max_e J(e)}{\min_{e'} J(e')}.$$

Let $L = \{0, \ldots, \lfloor \log \Delta \rfloor\}$ and $M = \{0, \ldots, \lfloor \log \Gamma \rfloor\}$. Let $j_{min} = \min_{e \in E} J(e)$. Without loss of generality, we assume that $\min_{e \in E}\{\ell(e)\} = 1$. We define $H_k^i = \{e \in E : \ell(e) \in [2^i, 2^{i+1}) \wedge J(e) \in [j_{min} \cdot 2^k, j_{min} \cdot 2^{k+1})\}$, for $i \in L, k \in M$. Note that for edge $e \in E$, its power level $J(e)$ is given as an input, and so our assumption of the AWGN model (equation 2) implies that its capacity $cap(e)$ is also fixed. We will be interested in two special cases of power level choices: (i) *Uniform power levels*, in which we have $J(e) = J$ for a constant J - in such a case, we denote the input instance as $\mathcal{I} = (V, E, \mathcal{D}, J)$, and (ii) *Linear power levels*, in which $J(e) = c_1 \cdot \ell(e)^\alpha$ where c_1 is a constant and α is the path-loss exponent as discussed earlier. As suggested in [23], we assume $\alpha > 2$.

Note that for any case of power level choice, for any edge $e = (u, v) \in E$, we require $J(e) \geq (1 + \epsilon)\beta N_0 \ell(e)^\alpha$ for the transmission on edge e to be feasible, even in the absence of any other interference, where ϵ is a small positive slack. We compare our solution with that obtained by the optimal solution with the criteria that optimum solution uses power level $J(e)$ that satisfies the following condition: $J(e) \geq \beta N_0 \ell(e)^\alpha \forall e \in E$.

4.1 Interference Model

We use the SINR model of interference as described in [7, 23]. In this setting, a given set $E' = \{e_i = (u_i, v_i) : i = 1, \ldots, k\}$ of links can simultaneously communicate successfully, if for each $e_i = (u_i, v_i) \in E'$, the $SINR(v_i)$ satisfies Equation 1. For any edge $e_i = (u_i, v_i) \in E'$, we define $I_r(v_i, E') = \sum_{e_j=(u_j,v_j)\neq e_i} \frac{J(e_j)}{d(u_j,v_i)^\alpha}$ as the the interference at receiver v_i due to all other transmissions - we will simply denote this as $I_r(v_i)$ if the set E' is clear from the context.

4.2 Link rates and feasible end-to-end schedules

We consider a set $\mathcal{D} = \{1, \ldots, k\}$ of connections, with s_i and t_i denoting the source and destination respectively, for connection i. Let $f_i(e)$ denote the mean flow rate on link e for the ith connection, and let $f(e) = \sum_i^k f_i(e)$ denote the total link flow. We let $x(e) = f(e)/cap(e)$ denote the *link utilization* - this denotes the fraction of time link e is used. The vectors $\bar{f}$ and $\bar{x}$ are called the flow and *link utilization* vectors respectively. An end-to-end schedule $\mathcal{S}$ describes the specific times at which packets are transmitted over the links of the network. For schedule $\mathcal{S}$, let $X(e, t)$ be an indicator variable such that

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

We say that $\mathcal{S}$ is valid if the SINR constraints are satisfied at all the receivers at every time t. We say that $\mathcal{S}$ feasibly schedules the *link utilization* vector $\bar{x}$ if we have $\lim_{T\to\infty} \sum_{t\leq T} \frac{X(e,t)}{T} = x(e)$ for each edge e - in this case, we say that $\mathcal{S}$ corresponds to the utilization vector $\bar{x}$. The rate region $\mathcal{X}(\mathcal{I})$ is the space of all utilization vectors $\bar{x}$ for the instance $\mathcal{I}$ of TM-SINR that can be scheduled feasibly.

Let r_i denote the end-to-end rate on the ith connection in bits per second, resulting from the flow vector $\bar{f}$. In this paper we are interested in maximizing the total end-to-end rate $\sum_i^k r_i$. For an instance $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ of TM-SINR, let $r_{opt}(\mathcal{I})$ denote the maximum possible total throughput rate that is feasible. We say that a utilization vector $\bar{x} \in \mathcal{X}(\mathcal{I})$ is a γ-approximation to the throughput maximization problem if the resulting total rate achieved is at least $\gamma \cdot r_{opt}(\mathcal{I})$; we say that an algorithm is a γ-approximation algorithm, if for any instance $\mathcal{I}$ of TM-SINR, it provably produces such a γ-approximate solution $\bar{x} \in \mathcal{X}(\mathcal{I})$ in *polynomial time*. Note that this is a worst case approximation result.

4.3 Congestion Measure

Following [7], we define a notion of congestion C, that will play a key role in our algorithm. For $e = (u, v) \in E$, let

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

and let $C = \max_{e\in E'} |C(e, E')|$. Here, a is a constant such that $a \geq 4 \sqrt[\alpha]{\frac{48\beta(1+\epsilon)}{\epsilon(\alpha-2)}}$, ϵ is a small positive slack and $\alpha > 2$ is the path-loss exponent. The significance of the congestion C is that it provides a lower bound on the number of feasible simultaneous transmissions [7], which we use to approximate r_{opt}. Figure 1 demonstrates the

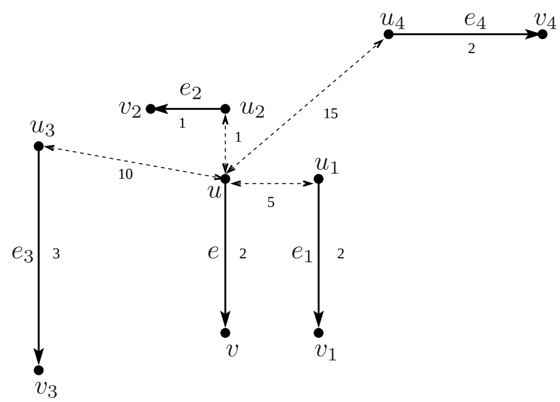


Figure 1: Illustrating congestion measure for a link. The solid lines represent edges along with edge lengths (e.g., $l(e) = 2$). The dotted lines represent distance between two nodes (e.g., $d(u, u_1) = 5$). Let $a = 6$. Then $C(e) = \{e, e_1, e_3\}$ for link $e = (u, v)$. By definition, $e_4, e_2 \notin C(e)$ since $a \cdot \ell(e_4) < d(u, u_4)$ and $\ell(e_2) < \ell(e)$.

congestion measure for link $e = (u, v)$. Intuitively, any link $e = (u, v)$ with high congestion value could have the SINR constraints violated at its receiver v.

Table 1 gives a list of most of the notation used in this paper.

5 SINR vs Graph-based models

In this section, we compare the SINR and graph-based models in the context of the throughput maximization problem. Given an instance $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ of TM-SINR, we follow the approach of [23] in constructing an "equivalent" connectivity graph $G = (V, E_{gm})$ and a resulting instance $\mathcal{I}_{gm}$ in a graph-based model in the following manner. Recall the notation from Section 4. In the rest of this section, we will consider instances $\mathcal{I}$ of TM-SINR in which every node $u \in V$ uses a fixed power level $J(u) = J$ for every incident link $e = (u, v) \in E$. We associate a transmission range of $r(u) = (J(u)/c_1)^{1/\alpha}$ with every node $u \in V$ for a constant c_1, giving rise to a disk graph $G = (V, E_{gm})$ with $(u, v) \in E_{gm}$ if $d(u, v) \leq r(u)$. This is a directed graph in general, if nodes have non-uniform transmission ranges. The corresponding instance $\mathcal{I}_{gm}$ consists of this graph G along with the same set $\mathcal{D}$ of connections, as in $\mathcal{I}$. Note that the set of edges on which transmissions can happen is the same in both models. For every edge $e \in E_{gm}$, we use the same expression for $cap(e)$, the capacity of edge e as in $\mathcal{I}$, since this comes from the AWGN model. What is different is the interference - we can now use any graph-based interference model to specify the set $I_{gm}(e)$ of edges that interfere with e - for concreteness, we use the distance-2 matching model [17], which defines $I_{gm}(e) = \{e' = (u', v') : d_G(\{u, v\}, \{u', v'\}) \leq 1\}$, where $d_G()$ defines the distance between two sets in the graph G. A schedule is valid in the graph-based model, if at any time, no edge e is simultaneously scheduled along with some edge $e' \in I_{gm}(e)$. Let $r_{opt}^{gm}(\mathcal{I}_{gm})$ denote optimum throughput rate possible for this instance in the graph-based model.

We show the following results in this section.

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

The above results show that if the power levels are fixed, the total throughput in both the models is very different - this is in contrast to the results of [21, 23], which show that by choosing appropriate power levels, a much higher throughput is possible in the SINR model for the same instance.

n	number of nodes.
V	set of nodes.
$E = \{(u,v) \in V \times V\}$	set of all edges.
N_{out}	set of outgoing edges from node u.
N_{in}	set of incoming edges to node u.
$\mathcal{D} = \{(s_1,t_1),..,(s_k,t_k)\}$	set of source-destination pairs.
$f_i(e)$	flow on edge e for ith connection.
$f(e) = \sum_{i=1}^{k} f_i(e)$	total flow on edge e.
$r_i = \sum_{e=(s_i,v)} f_i(e)$	rate for flow i .
$r = \sum_{i}^{k} r_i$	total rate.
$cap(e)$	capacity of link e.
$x(e) = \frac{f(e)}{cap(e)}$	link utilization
W	single window or frame.
$C(e)$	congestion set for link e.
$C = \max_{e \in E} \lvert C(e) \rvert$	Max congestion.
$\bar{J}$	power level vector.
$j_{min} = \min_{e \in E} J(e)$	minimum power level.
Δ	ratio of max. to min. node-distances.
Γ	ratio of max. to min. power assigned.
$L = \{0,\ldots,\lfloor \log \Delta \rfloor\}$	set of possible edge lengths.
$M = \{0,\ldots,\lfloor \log \Gamma \rfloor\}$	set of possible power values.
$\mathcal{S}$	a valid schedule.
α	path-loss exponent, assumed to be ¿ 2.
β	antenna gain.
N_0	ambient noise.
$a, \lambda_0, \lambda_1, \lambda_2, \epsilon, c_1$	positive constants.

Table 1: Notation used in this paper

5.1 Uniform power levels

We construct the following instance $\mathcal{I} = (V, E, \mathcal{D}, J)$ of TM-SINR, with *uniform* power level J for all transmissions. Let $R = (J/c_1)^{1/\alpha}$ be the corresponding transmission range in the corresponding graph model, as discussed earlier; we assume that R is a large integer. Let $V = \{u_0\} \cup_{i=1}^{n} \{u_i, v_i\}$ be a set of nodes, which are placed in the following manner. Imagine a circle of radius $R/2$ centered at node u_0, and the nodes $u_1, \ldots, u_n$ uniformly placed on the circumference of this circle at a spacing of $\Theta(\sqrt{R})$, so that $n = \Theta(\sqrt{R}) = c_2 \cdot \sqrt{R}$ (cf. Figure 2). Each v_i is at a unit distance from u_i, for $i = 1, \ldots, n$. Let the connections in $\mathcal{D}$ in the instance $\mathcal{I}$ be all the pairs $e_i = (u_i, v_i)$, for all $i = 1, \ldots, n$. Let $cap = cap(e_i)$ denote the capacity of any link e_i in bits/sec; note that this is the same for every edge e_i in this setting. For simplicity, we ignore the ambient noise, i.e., assume $N_0 = 0$. It is easy to extend these results to take the noise into account.

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

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

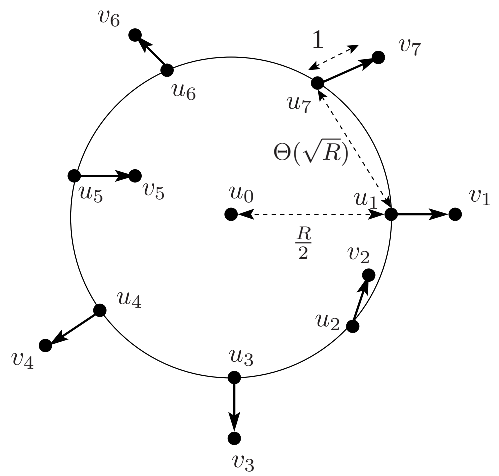


Figure 2: Illustrating an example that compares the throughput achieved between SINR and graph- based models for *uniform* power levels. A circle of radius $R/2$ is centered at node u_0. Distance between any two adjacent u_i, u_j is $\Theta(\sqrt{R})$. Each edge $e_i = (u_i, v_i)$ has length $\ell(e_i)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

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

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

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

5.2 Linear Power Levels

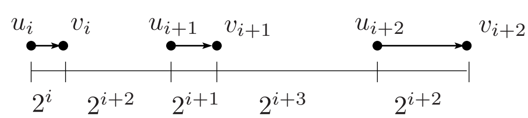


Figure 3: Illustrating an example that compares the throughput achieved between SINR and graph- based models for *linear* power levels. Representing a line topology, such that each edge $e_i = (u_i, v_i)$ has length $\ell(e_i) = 2^i$ and $d(v_i, u_{i+1}) = 2^{i+2}$. $J(e_i) = c_1 \ell(e_i)^\alpha$, for constant c_1.

We now construct an instance $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ of TM-SINR with *linear* power levels, i.e., for each $e \in E$, $J(e) = c_1 \ell(e)^\alpha$, for constant c_1. The set $V = \cup_{i=1}^{n} \{u_i, v_i\}$ has $2n$ nodes, which are located on a line in the order $u_1, v_1, u_2, v_2, \ldots, u_n, v_n$. For all $i = 1, \ldots, n$, we have $d(u_i, v_i) = R_i = 2^i$, and for all $i = 1, \ldots, n-1$, we have $d(v_i, u_{i+1}) = 2^{i+2}$ (cf. Figure 3). The set $E = \{e_i = (u_i, v_i) : i = 1, \ldots, n\}$ will be the only edges used for transmission, with $J(e_i) = c_1 R_i^\alpha$, for each i. All the connections in $\mathcal{D}$ in this instance are the pairs $e_i = (u_i, v_i)$, for

$i = 1, \ldots, n$. Because of our AWGN model for the link capacities, as discussed in Equation 2, it follows that for all $e_i \in E$, $cap(e_i) = cap$ is a fixed value. Each node u_i has only one incident edge in the set E, so for the graph-based model, we set $r(u_i) = R_i = 2^i$. Therefore, for the corresponding graph-based instance $\mathcal{I}_{gm}$, the connectivity graph $G = (V, E_{gm})$ has $E_{gm} = \{e_i : i = 1, \ldots, n\}$.

Lemma 2 *For the instance $\mathcal{I}$ and the corresponding graph-based instance $\mathcal{I}_{gm}$ described above, we have $r^{gm}_{opt}(\mathcal{I}_{gm})/r_{opt}(\mathcal{I}) = \Theta(n)$.*

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

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

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

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

6 Throughput Maximization with SINR constraints TM-SINR

In this Section, we consider the generic case of power levels, wherein the power level on every edge $e \in E$ is $J(e)$ and is specified by the corresponding vector $\bar{J}$. We first formulate a linear program for the TM-SINR problem and then derive the necessary and sufficient conditions for link flow stability.

6.1 Problem Formulation

In this Section we mathematically formulate the TM-SINR problem. We consider input instances of TM-SINR specified as $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$. Recall the notation from section 4. It can be seen that due to the non-linearity of the SINR constraints, the exact formulation of the TM-SINR problem is *non-convex*. We develop a linear programming *relaxation* of this problem by combining the approaches of [7, 17] - we show that both **necessary** and **sufficient** conditions can be derived for the feasible rate region by considering the total *link utilization* in the edges in the set $C(e)$ for any edge e. In order to achieve a *stable* and *feasible* schedule, we partition the set E of edges into sets $H^i_k = \{e = (u, v) \in E : \ell(e) \in [2^i, 2^{i+1}) \wedge\ J(e) \in [j_{min} \cdot 2^k, j_{min} \cdot 2^{k+1})\}, \forall i \in L, k \in M\}$. As we discuss in the next sub-sections, this partitioning helps the scheduling algorithm to bound the number of links that can be scheduled simultaneously without violating the SINR constraints at every receiver. Our formulation for instance $\mathcal{I}$ described below is denoted by $\mathcal{P}(\lambda, \mathcal{I})$, where λ is a parameter.

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

$$\forall i \in \mathcal{D}, r_i = \sum_{e \in N_{out}(s_i)} f_i(e) \tag{3}$$

$$\forall i \in \mathcal{D}, \sum_{e \in N_{in}(s_i)} f_i(e) = 0 \tag{4}$$

$$\forall e \in E,\ x(e) = \sum_{i \in \mathcal{D}} f_i(e)/cap(e) \tag{5}$$

$$\forall i \in \mathcal{D}, \forall u \neq s_i, t_i, \quad \sum_{e \in N_{out}(u)} f_i(e) = \sum_{e \in N_{in}(u)} f_i(e) \tag{6}$$

$$\forall e \in E, \forall i \in L, \forall k \in M, \quad \sum_{e' \in C(e) \cap H_k^i} x(e') \leq \lambda \tag{7}$$

In the above formulation, constraints (3) define the total rate r_i for each connection, constraints (4) define the link utilization $x(e)$ for each link e, constraints (5, 6) ensure flow conservation, and constraints (7) are relaxed congestion constraints - these are the *key* constraints that allow us to use this program to derive upper and lower bounds on the optimum rate. The program $\mathcal{P}(\lambda, \mathcal{I})$ has *polynomial* size and can be solved in *polynomial time*.

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

6.2 Link-Flow Scheduling: Necessary Conditions

The following Lemma shows that $\mathcal{P}(\lambda, \mathcal{I})$ gives an upper bound on $r_{opt}(\mathcal{I})$ for a suitable choice of λ.

Lemma 3 *Let $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ be an instance of the* TM-SINR *problem, and let $\bar{x} \in \mathcal{X}(\mathcal{I})$ be any* feasible link utilization *vector. Then, $\bar{x}$ satisfies the following conditions:*

$$\forall e \in E, \forall i \in L, \forall k \in M, \quad \sum_{e' \in C(e) \cap H_k^i} x(e') \leq \lambda_0,$$

where $\lambda_0 = \theta \frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2$, and a is the constant defined in Section 4.3. This implies that $\bar{x}$ is a feasible solution to the program $\mathcal{P}(\lambda_0, \mathcal{I})$.

Proof Since the *link utilization* vector $\bar{x}$ is feasible, there exists a *stable* schedule $\mathcal{S}$ which achieves the link rates specified by $\bar{x}$. Recall the notation $X(e,t)$ from Section 4. Let $E_t = \{e : X(e,t) = 1\}$ denote the set of links that transmit at time t in this schedule. We now focus on any edge $e = (u,v) \in E_t$. Let $A_t(e) = E_t \cap C(e) = \{e_j = (u_j, v_j) \in C(e) : j = 1, \ldots, s\}$ be a set of links in $C(e)$ that are scheduled simultaneously at time t. Define $G_k = \{e \in E : j_{min} \cdot \theta^k \leq J(e) < j_{min} \cdot \theta^k\}, \forall k \in M$. We argue below that the number of links that can be simultaneously scheduled from set $Q_{t,k}(e) = A_t(e) \cap G_k$, for any edge e, at any time t and any $k \in M$ is $O(1)$. Let the links in the set $Q_{t,k}(e)$ be numbered in non-decreasing order of their lengths, so that $\ell(u_1, v_1) \leq \ell(u_2, v_2) \leq \ldots \leq \ell(u_c, v_c)$ (cf. Figure 4). For simultaneously successful transmission of these links, the SINR at each node v_j, and in particular at node v_c needs to be at least β.

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

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

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

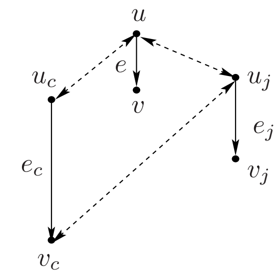


Figure 4: For a given link $e = (u, v)$ and set $Q_{t,k}(e)$, $d(u_j, v_c) \le (2a + 1)d(u_c, v_c)$, where $e_c, e_j \in C(e)$ and e_c is the link with longest length in set $Q_{t,k}(e)$.

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

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

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

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

Rearranging, we have

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

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

$$\forall e \in E, \forall t, \sum_{e'\in Q_{t,k}(e)} X(e', t) \le \lambda_0.$$

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

$$\forall e \in E, \forall k \in M, \forall i \in L, \sum_{e'\in C(e)\cap H_k^i} \sum_{t\le T} X(e', t) \le T\lambda_0. \tag{8}$$

Dividing both sides of (8) by T, the Lemma follows from the definition of $x(e)$ in Section 4.2. ■

6.3 Link-Flow Scheduling: sufficient conditions

In this section, we show that the program $\mathcal{P}(\lambda, \mathcal{I} = (V, E, \mathcal{D}, \bar{J}))$ can be used to derive sufficient conditions for link flow stability for the instance $\mathcal{I}$ of TM-SINR, for a suitable value of the parameter λ. This requires showing that a solution $\bar{x}$ to the program $\mathcal{P}(\lambda, \mathcal{I})$ can be scheduled feasibly, under suitable conditions on λ and $\bar{J}$. We describe an algorithm FrameSchedule for constructing a feasible schedule below.

We assume that time is divided into sufficiently large frames, with the length of each frame $|W| = w$. Recall the definitions of Δ, Γ and the sets H_k^i from Section 4. We further subdivide each frame W into $(1+\log\Delta)\cdot(1+\log\Gamma)$ sub-frames $W_{i,k}$, such that $|W_{i,k}| = w' = w/(1+\log\Delta)\cdot(1+\log\Gamma)$, $\forall i \in L, \forall k \in M$. We assume that lengths of sub-frames $W_{i,k}, \forall i \in L, k \in M$ and that $x(e)\cdot w'$ is integral for all $e \in E$. Algorithm FrameSchedule constructs a periodic schedule $\mathcal{S}$ by repeating a schedule $\mathcal{S}_W$ for every frame W. Within each sub-frame $W_{i,k}$, the algorithm considers only the edges from the set H_k^i and assigns $s(e) = x(e)\cdot w'$ slots for each edge $e \in H_k^i$ such via a greedy coloring scheme. In this scheme we consider the edges in a decreasing order of their lengths and assign time slots to each edge such that no two edges e, e' with $e \in C(e')$ are assigned the same time slot. The final schedule is constructed by combining schedules $\mathcal{S}_W$ for all the frames.

Algorithm 1: FrameSchedule

Input : (i) E,(ii)$\bar{x}$,(iii)W,(iv)$\bar{J}$,(v)w
Output: Sets $s(e)$ for all edges e and schedule $\mathcal{S}_W$

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

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

Lemma 4 *Algorithm FrameSchedule correctly assigns $|s(e)| = x(e)\cdot w'$ slots for each edge e, if the* link utilization *vector $\bar{x}$ satisfies the following conditions $\forall e \in E, \forall i \in L, \forall k \in M$:*

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

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

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

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

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

Since the program $\mathcal{P}(\lambda, \mathcal{I})$ has size polynomial in n, the link utilization rates $x(e)$ are rational and of the form Z_1/Z_2, with both $Z_1, Z_2 \le 2^{n^c}$, for some constant c. Therefore, the frame size w is also bounded by 2^{n^c}. As we discuss later in Section 6.4, we can modify the link utilization vector $\bar{x}$ so that the frame size w becomes a polynomial in n, with a slight reduction in the total achievable throughput.

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

Lemma 5 *Let $\bar{x}$ be a feasible solution to the program $\mathcal{P}(1/(1+\log\Delta)(1+\log\Gamma), \mathcal{I}=(V,E,\mathcal{D},\bar{J}))$. Then, Algorithm* FrameSchedule *produces a feasible schedule corresponding to $\bar{x}$ for the instance $\mathcal{I}'=(V,E',\mathcal{D},(1+\epsilon)\bar{J}$ of* TM-SINR*, where $E'=\{e\in E: J(e)\geq(1+\epsilon)\beta N_0\ell(e)^\alpha\}$ in which the SINR constraints are satisfied at all receivers, for constants $a\geq 2\sqrt[\alpha]{\frac{48\theta\beta(1+\epsilon)}{\epsilon(\alpha-2)}}$, $\alpha>2,\epsilon>0$, and $\theta=2$.*

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

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

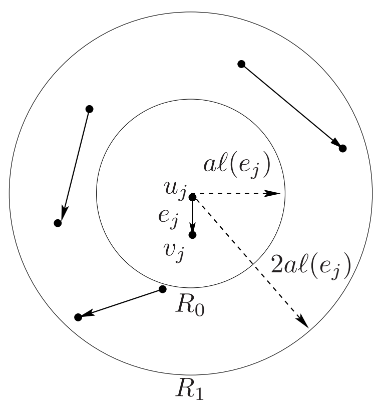


Figure 5: For a given link $e_j=(u_j,v_j)\in E_t$, construct rings of radius $a\ell(e_j)$ around u_j. We calculate the interference experienced by node v_j due to other simultaneously transmitting links.

Consider any $e_j=(u_j,v_j)\in E_t$. We estimate the SINR at v_j in the following manner. As in [7,23], we partition the plane into rings R_d centered at u_j (cf. Figure 5) for $d=0,1,\ldots$, each of width $a\ell(e_j)$ around u_j. Each ring R_d consists of all links $e_m=(u_m,v_m)$, for which $da\ell(e_j)\leq d(u_j,u_m)<(d+1)a\ell(e_j)$. As derived earlier, for any $e_m\neq e_j$, we have $d(u_j,u_m)>a\max\{\ell(e_j),\ell(e_m)\}$, which implies R_0 does not contain any links in E_t other than e_j. The area of the ring R_d can be calculated as,

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

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

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

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

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

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

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

Therefore the SINR at receiver v_j is at least

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

which is at least β if $a^\alpha \geq 2^\alpha \frac{48\theta\beta(\epsilon+1)}{\epsilon(\alpha-2)}$ and $J(e_j) \geq (1+\epsilon)\beta N_0 \ell(e_j)^\alpha$. We therefore choose a (defined in Section 4.3) that satisfies the above condition. In order guarantee the feasibility and validity of the schedule $\mathcal{S}$, we add a small slack ϵ to the power levels and enforce the following constraints: $\forall e \in E, J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha$. This corresponds to a *different* instance $\mathcal{I}' = (V, E', \mathcal{D}, (1+\epsilon)\bar{J})$ of TM-SINR, where $E' = \{e \in E : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$. The vector $\bar{x}$ produced by algorithm FrameSchedule is therefore not valid for the original instance $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$, but is valid for the instance $\mathcal{I}'$ of TM-SINR. ■

6.4 Putting everything together

For an input instance $\mathcal{I}' = (V, E', \mathcal{D}, (1+\epsilon)\bar{J})$ of TM-SINR, we first compute the optimum *link utilization* vector $\bar{x}$ by solving the linear program $\mathcal{P}(1/(1+\log\Delta)\cdot(1+\log\Gamma), \mathcal{I} = (V, E, \mathcal{D}, \bar{J})$, where $E' = \{e \in E : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$. We know from Lemma 5 that $\bar{x}$ can be scheduled feasibly for the instance $\mathcal{I}'$. The following theorem shows that the rate achieved by $\bar{x}$ is within a provable factor of $r_{opt}(\mathcal{I})$ - thus, this is a **bi-criteria** approximation, in which we compare the quality of the solution produced by our algorithm with respect to the optimum for an instance that uses slightly less power.

Theorem 1 *Let $\mathcal{I}' = (V, E', \mathcal{D}, (1+\epsilon)\bar{J})$ be an instance of* TM-SINR*, and let $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ be the corresponding instance for which the optimum rate $r_{opt}(\mathcal{I})$ is considered, such that $E' = \{e \in E : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$, for any $\epsilon > 0$. The optimum solution $\bar{x}$ to the program $\mathcal{P}(1/(1+\log\Delta)\cdot(1+\log\Gamma), \mathcal{I})$ yields a feasible and stable* link utilization *vector for the instance $\mathcal{I}'$, and results in a total throughput of at least $\Omega(r_{opt}(\mathcal{I})/\lambda_0(1+\log\Delta)\cdot(1+\log\Gamma))$, for $\lambda_0 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2$, and a as defined in Section 4.3.*

Proof Let $\bar{x}_{opt}$ be the optimum utilization vector for the instance $\mathcal{I}$ of TM-SINR, achieving a total throughput rate of $r_{opt}(\mathcal{I})$. From Lemma 3, it follows that $\bar{x}_{opt}$ is a feasible solution if it satisfies the conditions stated in Lemma 3 and hence is a feasible solution to the program $\mathcal{P}(\lambda_0, \mathcal{I})$, for $\lambda_0 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2$, and a as defined in Section 4.3. We now scale down the *link utilization* vector $\bar{x}_{opt}$ to achieve a new vector $\bar{y}$ such that $\bar{y} = \frac{1}{\lambda_0(1+\log\Delta)\cdot(1+\log\Gamma)}\bar{x}_{opt}$. Since $\mathcal{P}(\lambda_0, \mathcal{I})$ is a linear program, it follows that $\bar{y}$ is a feasible solution to the program $\mathcal{P}(1/(1+\log\Delta)\cdot(1+\log\Gamma), \mathcal{I})$, and results in a total throughput rate of $\frac{r_{opt}(\mathcal{I})}{\lambda_0(1+\log\Delta)\cdot(1+\log\Gamma)}$. This implies that the optimum solution $\bar{x}$ to the program $\mathcal{P}(1/(1+\log\Delta)\cdot(1+\log\Gamma),\mathcal{I})$ also results in a total throughput rate of at least $\frac{r_{opt}(\mathcal{I})}{\lambda_0(1+\log\Delta)\cdot(1+\log\Gamma)}$. Finally, by Lemma 5, it follows that $\bar{x}$ can be scheduled feasibly for the instance $\mathcal{I}'$ of TM-SINR. Therefore, the theorem follows. ■

Our presentation of the LP based technique and Algorithm FrameSchedule assume a centralized and synchronized setting, with traffic arrivals at a constant bit rate. Further, the running time for Algorithm FrameSchedule depends on the frame size w, which need not be polynomial. However, as we discuss below, the rate vector obtained from the LP can be modified slightly so that the frame size becomes polynomial, with a slight reduction in total throughput. Implementing the LP and the scheduling in a distributed are challenging problems, but approaches of [3] and [12] for distributed flow computation and random access scheduling, respectively, could be useful. One difference, though, is that these papers are not based on the SINR model.

Modifying the rate vector to ensure polynomial sized frame. We now describe a simple idea to modify the rate vector $\bar{x}$ so that the frame size becomes polynomial sized, so that the running time of Algorithm FrameSchedule becomes polynomial. This involves the following steps.

1. For each edge e, round $x(e)$ to the nearest multiple of $1/n^c$ for a constant c. In other words, we consider $x'(e) = \frac{\lfloor x(e)n^c \rfloor}{n^c}$.
2. Next, consider a maximum multi-commodity path flow $\bar{f}'$ with capacities $x'(e)cap(e)$. If the capacities are scaled by a factor of n^c, all these capacities are integral (and at most a polynomial) because of the previous step. Therefore, $f'(e)\cdot n^c$ is also integral for each e, using standard properties of network flows. Let $\bar{x''} = \sum_i f_i'(e)/cap(e)$ be the resulting link utilization for the flow $f'()$.
3. Run Algorithm FrameSchedule with the link utilization vector $\bar{x''}$, for which the frame size w will be at most n^c.

It is easy to see that the total throughput resulting from the scaled rate vector $\bar{f}'$ is at least $1-1/n^c$ times the original, and therefore, the statement of Theorem 1 still holds, but the scheduling step now runs in polynomial time.

7 Throughput Maximization for Uniform Power Levels

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

7.1 Problem Formulation

We consider input instances of TM-SINR specified as $\mathcal{I} = (V, E, \mathcal{D}, J)$, with *uniform* power level of $J(e) = J$ for every edge $e \in E$. The problem formulation for the TM-SINR problem for *uniform* power levels is similar to the one presented in Section 6.1. Recall that in Section 6.1, we partitioned the set E of edges into sets H_k^i based on the edge lengths and power levels, (i.e. $H_k^i = \{e = (u,v) \in E : \ell(e) \in [2^i, 2^{i+1}) \wedge J(e) \in [j_{min}\cdot 2^k, j_{min}\cdot 2^{k+1})\}, \forall i \in L, k \in M$). Since the power levels are *uniform*, we only need to partition the set E of edges based on edge lengths. Therefore we obtain sets $H^i = \{e = (u,v) \in E : \ell(e) \in [2^i, 2^{i+1})\}, \forall i \in L$. For an instance $\mathcal{I} = (V, E, \mathcal{D}, J)$ of TM-SINR, we define a different formulation $\mathcal{P}_u(\lambda,\mathcal{I})$ by replacing the constraints (7) in the program $\mathcal{P}(\lambda,\mathcal{I})$ by the constraints

$$\forall e \in E, \forall i \in L \sum_{e' \in C(e)\cap H^i} x(e') \leq \lambda. \tag{9}$$

7.2 Link-Flow Scheduling: Necessary Conditions

Lemma 6 *Let $\mathcal{I} = (V, E, \mathcal{D}, J)$ be an instance of the* TM-SINR *problem with* uniform *power level J, and let $\bar{x} \in \mathcal{X}(\mathcal{I})$ be any feasible* link utilization *vector. Then, $\bar{x}$ satisfies the following conditions:*

$$\forall e \in E, \forall i \in L, \sum_{e' \in C(e) \cap H^i} x(e') \leq \lambda_1,$$

where $\lambda_1 = \theta \frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 1$, and a is the constant defined in Section 4.3. This implies that $\bar{x}$ is a feasible solution to the program $\mathcal{P}_u(\lambda_1, \mathcal{I})$.

Proof The proof is similar to that of Lemma 3. Recall the notations used in the proof of Lemma 3. For the case of uniform power levels, we only consider the sets $H^i \forall i \in L$. We further do not consider the set G_k and set $Q_t(e) = A_t(e)$. By following the sequence of steps used in Lemma 3 and substituting $\theta = 1$ the Lemma follows. ■

7.3 Link-Flow Scheduling: Sufficient Conditions

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

Algorithm 2: UniformFrameSchedule

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

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

Lemma 7 *Algorithm UniformFrameSchedule correctly assigns $|s(e)| = x(e) \cdot w'$ slots for each edge e, if the* link utilization *vector $\bar{x}$ is any feasible solution to the program $\mathcal{P}(\frac{1}{(1+\log \Delta)}, \mathcal{I})$.*

Proof The proof is similar to that of Lemma 4. Since we have *uniform* power levels, $\log \Gamma = 0$. Further we only consider sets H^i,$\forall i \in L$. By making these modifications to Lemma 4, it follows that algorithm UniformFrameSchedule indeed assigns $x(e) \cdot w'$ slots for each edge e. We now derive the conditions under which the schedule is valid. ■

Lemma 8 *Let $\bar{x}$ be a feasible solution to the program $\mathcal{P}_u(1/(1+\log\Delta), \mathcal{I} = (V, E, \mathcal{D}, \bar{J}))$, Then, Algorithm* UniformFrameSchedule *produces a feasible schedule corresponding to $\bar{x}$ for the instance $\mathcal{I}' = (V, E', \mathcal{D}, (1+\epsilon)\bar{J}$ of* TM-SINR*, where $E' = \{e \in E : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$ in which the SINR constraints are satisfied at all receivers, for constants $a \geq 2\sqrt[\alpha]{\frac{48\theta\beta(1+\epsilon)}{\epsilon(\alpha-2)}}$, $\alpha > 2, \epsilon > 0$, and $\theta = 1$.*

Proof It can be seen that by considering only sets $H^i, \forall i \in L$ and by substituting $\log\Gamma = 0, \theta = 1$, in the proof of Lemma 5, we can prove the above Lemma. ■

Theorem 2 *Let $\mathcal{I}' = (V, E', \mathcal{D}, (1+\epsilon)\bar{J})$ be an instance of* TM-SINR*, and let $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ be the corresponding instance for which the optimum rate $r_{opt}(\mathcal{I})$ is considered, such that $E' = \{e \in E : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$, for any $\epsilon > 0$. The optimum solution $\bar{x}$ to the program $\mathcal{P}_u(1/(1+\log\Delta), \mathcal{I})$ yields a feasible and stable* link utilization *vector for the instance $\mathcal{I}'$, and results in a total throughput of at least $\Omega(r_{opt}(\mathcal{I})/\lambda_1(1+\log\Delta))$, for $\lambda_1 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 1$, and a as defined in Section 4.3.*

Proof The proof of Theorem 1 can be applied here by substituting $\log\Gamma = 0$, λ_0 with λ_1 and program $\mathcal{P}$ with $\mathcal{P}_u$. ■

8 Improved Approximations for Linear Power Levels

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

Let $\bar{J}$ be the power value vector with $J(e) = c_1\ell(e)^\alpha, \forall e \in E$. Recall the definition of the sets H^i from Section 7.1. In order to get a better approximation, we partition the set of edges E into sets H^i based on their lengths. It can be seen that $\forall e', e'' \in H^i, J(e')/2^\alpha \leq J(e'') \leq J(e'), \forall i \in L$. For an instance $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ of TM-SINR, we consider the program $\mathcal{P}_u(\lambda, \mathcal{I})$ (described in Section 7.1) instead of the program $\mathcal{P}(\lambda, \mathcal{I})$.

Lemma 9 *Let $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ be an instance of the* TM-SINR *problem with* linear *power levels $\bar{J}$, and let $\bar{x} \in \mathcal{X}(\mathcal{I})$ be any feasible* link utilization *vector. Then, $\bar{x}$ satisfies the following conditions:*

$$\forall e \in E, \forall i \in L, \sum_{e' \in C(e) \cap H^i} x(e') \leq \lambda_2,$$

where $\lambda_2 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2^\alpha$, and a is the constant defined in Section 4.3. This implies that $barx$ is a feasible solution to the program $\mathcal{P}_u(\lambda_2, \mathcal{I})$,

Proof The proof of Lemma 6 can be applied here, by substituting $\theta = 2^\alpha$. ■

Theorem 3 *Let $\mathcal{I}' = (V, E', \mathcal{D}, (1+\epsilon)\bar{J})$ be an instance of* TM-SINR*, and let $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ be the corresponding instance for which the optimum rate $r_{opt}(\mathcal{I})$ is considered, such that $E' = \{e \in E : J(e) \geq (1+\epsilon)\beta N_0 \ell(e)^\alpha\}$, for any $\epsilon > 0$. The optimum solution $\bar{x}$ to the program $\mathcal{P}_u(1/(1+\log\Delta), \mathcal{I})$ yields a feasible and stable* link utilization *vector for the instance $\mathcal{I}'$, and results in a total throughput of at least $\Omega(r_{opt}(\mathcal{I})/\lambda_2(1+\log\Delta))$, for $\lambda_2 = \theta\frac{(2a+1)^\alpha}{\beta} + 1$, $\theta = 2^\alpha$, and a as defined in Section 4.3.*

Proof The proof of is similar to that of Lemmas 7,8, and Theorem 2. We use algorithm UniformFrameSchedule to schedule the vector $\bar{x}$. By substituting $\theta = 2^\alpha$ and λ_1 with λ_2 in the proof of Lemma 8 and Theorem 2, the theorem follows. ■

9 Simulations

In this section, we conduct extensive simulations to validate the approximation techniques discussed in Section 6.3 and gain deeper insights into the theoretical model. Specifically, there are two main goals of our simulations: (i) validate the theoretical model and verify that the schedule produced by the greedy algorithm FrameSchedule (cf. Section 6.3) is feasible (i.e. SINR constraints are not violated at any given time), and (ii) compare the approximate solution with the optimal solution for a reasonable network setting and identify parameters that can allow us to improve the performance of our model. Our simulation setup is described below.

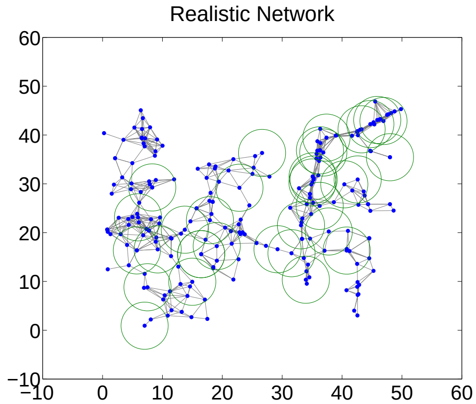
- **Network Type:** We consider a road traffic network corresponding to a distribution of 227 cars for a particular time instance in a region of downtown Portland, OR, obtained by running the TRANSIMS simulator [24]. We scaled down this network to fit in a $50m \times 50m$ region (cf. Figure 6(a))
- **Number of connections:** We experiment with varying number of connections, each of the source-destination pairs are chosen u.a.r. We denote these as k.
- **Edge Capacities:** All edges have a transmission rate of 5Mbps.
- **Transmission power:** We perform all experiments for the case of uniform power levels, where every transmitter can transmit at 40mW. The transmission range was set to 10m.
- **Number of seeds:** All data points are averaged over 5 runs of the experiment.

9.1 Validation of the theoretical model

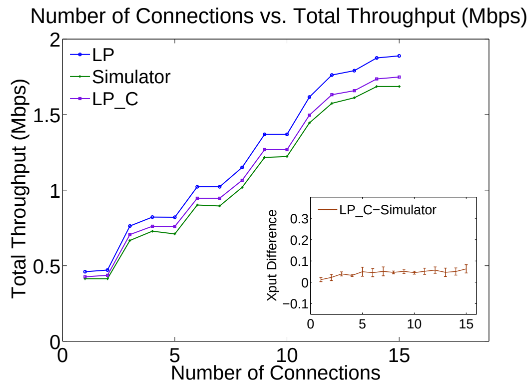
In Section 6.3 we have theoretically proved that the rate vectors obtained by solving the LP (cf. Section 6) can be feasibly scheduled without violating the SINR constraints. In this set of experiments, we verify the correctness of our theoretical model. **Goal:** The goal of this experiment is to verify the feasibility of the rates and schedule derived by the approximation algorithm for the TM-SINR problem in a realistic setting.

For a given network instance, we solve the LP using the Neos solver [27] and obtain the overall throughput and individual link rates. We then construct a centralized TDMA schedule using the greedy scheduling algorithm FrameSchedule and implement this schedule in the Qualnet simulator. The simulator decides if a packet has been successfully received or not by measuring the signal-to-noise ratio at every receiver and comparing it with the SINR threshold (β). We observe the overall throughput achieved by the simulator for a given schedule and compare this with the throughput obtained by solving the LP. We study the variation of throughput as a function of the number of connections. In this set of experiments, we do not consider the impact of routing; for a given set of randomly chosen source destination pair, we compute the shortest path using the Dijkstra's algorithm.
Results and Explanation: Figure 6(b) summarizes the results of our experiment averaged over 5 runs. We also plot the average difference between the LP and the simulator output for different number of connections and provide 95% confidence intervals according to the Gaussian distribution. We observe differences between the LP and the simulator output. These are mainly due to the delay introduced by transmission of control packets in the simulator. At the physical layer of the simulator, according to the 802.11 specification, a control packet (known as the PLCP preamble) is sent before transmitting a data packet. In the greedy scheduling algorithm (FrameSchedule), we divide the entire time frame into time-slots of equal lengths. In order to ensure that the packets sent at the start of every time-slot reach before or at the end of every time-slot, we set the duration of the time-slot to be an integral multiple of the transmission time. For example, if the transmission rate of every edge is 5Mbps, and the packet size is 1000 bytes, the transmission time is $1000 * 8/5 = 1600\mu$secs. The slot duration in this example would by $k \cdot 1600\mu secs$, where k is the number of packets sent in a given time-slot. The throughput in the scheduling algorithm is measured at the end of every time slot and is calculated as *total number of bytes received at the end of the time-slot/duration of the time-slot.* We do not consider the effects of propagation delay and the delay due to the transmission of control packets (control-delay). These delays are considered in the simulator. For successful transmission and reception of packets in any given time-slot, the simulator requires the duration of the time-slot to be slightly higher than that assumed in the greedy scheduling scheme. This causes the simulator throughput to be slightly different than the LP throughput. However, we can incorporate the effect of the control packets and the propagation delay in the LP formulation. The maximum link capacity of every link in the LP can be calculated as *packet size*$\cdot 8$*/transmission time + propagation*

delay + control delay. We call this LP as $LP_{compensated}$ and it can be seen from Figure 6(b), that the throughput achieved by $LP_{compensated}$ (denoted as LP_C in the plot) matches very closely with the simulator throughput. This shows that packet loss does not occur in the simulator due to the violations of the SINR constraints. We therefore conclude that (a) the greedy scheduling algorithm FrameSchedule is feasible, (b) the rates obtained by the LP are achievable in a realistic setting and, (c) effects of various delays such as propagation, control etc. can be incorporated in the LP.



(a) Realistic Network consisting of 227 nodes in a $50m \times 50m$ square area.



(b) Variation of throughput w.r.t. the number of connections as observed by the LP and the network simulator.

Figure 6: Network map and throughput

9.2 Comparison with the optimal solution

The necessary and sufficient conditions discussed in Sections 6.2, 6.3 conditions correspond to the upper and lower bounds on network capacity. Based on these bounds, we derive an approximation ratio that indicates how far the approximate solution can be from the optimal solution in the worst case. In this set of experiments, we aim to gain deeper insights into our approximation techniques and the bounds by performing a comparison with the optimal solution.
Goal: The goals of this experiment are to (a) observe how far the approximate solution is from the optimal solution for a reasonable network setting, (b) compare observed approximation ratio with theoretically estimated approximation ratio, (c) study the impact of different network types on performance of the approximation technique, and (d) identify parameters for improving the performance of the theoretical model.
For a given network instance, we solve the LP with the sufficiency conditions and calculate the optimal solution by performing an exhaustive search (brute-force technique). Since the brute-force technique is unlikely to scale for networks with multi-hop paths, we consider only one-hop source destination pairs. We also run the LP with the necessary conditions, in order to obtain an upper bound on the throughput. For a given network instance, we compute the *observed approximation ratio*, which is the ratio of the optimal solution and the LP solution and compare this with the theoretically derived worst-case approximation ratio, which is the ratio of the derived upper bound and the lower bound. For the case of uniform power levels and uniform link capacities this ratio is $\min\{k \cdot cap, \frac{(2a+1)^{\alpha}+\beta)(1+\log\Delta)}{\beta}\}$, where k denotes the number of connections, cap denotes the maximum transmission rate of every link and a, α, β, Δ are as defined in Section 4. We further compare the LP throughput with the one obtained by running the standard 802.11 random-access MAC protocol in the network simulator. We study the variation of overall throughput as a function of the number of connections.
Results and Explanation: Figure 7 plots the results of this experiment. We classify our observations in the following way:

- *Impact of length diversity:* For the case of non-uniform edge lengths the LP scales down the overall rates by a scaling factor of $1 + \log\Delta$, where Δ denotes the maximum inter-point separation (ratio of the maximum length edge and minimum length edge). Recall that in our approximation scheme, we partition the edges into

$\log \Delta$ bins based on their edge lengths, such that each bin $H^i = \{e \in L : \ell(e) \in [2^i, 2^{i+1})\}$. We then consider each bin separately and schedule all the links belonging to it (cf.Section 6.3). For the realistic network considered, the average length diversity was 6.15 suggesting the use of 3 bins. We observed that (cf. Figure 7(c)) this binning strategy highly underestimates the LP throughput. We therefore experimented with different bin-widths. By increasing the width of each bin by a quantity p, each bin can accommodate more links making $H^i = \{e \in L : \ell(e) \in [2^i, 2^{i+p})\}$. This results in fewer bins and hence, the throughput scaling factor can be reduced (the scaling factor is equal to the total number of bins). In order to ensure that the schedule is feasible by increasing the bin width, we impose additional conditions on the value of constant a (cf. Section 6.3). For the case of uniform power levels, the new constraints on the value of a are of the form $a \geq 2\sqrt[\alpha]{\frac{3 \cdot 2^{2p+2}\beta(1+\epsilon)}{\epsilon(\alpha-2)}}$, where p denotes the increase in the bin-width, $p = 1$ implies the original case. This condition can be derived by following the proof for Lemma 4. It should be noted that there is a tradeoff between p and a. Increasing the bin width reduces the scaling factor but increases the lower bound on the value of a. The interference set of every edge $C(e)$ is based on the value of a. As the value of a increases, more links can be included in the set $C(e) \forall e \in \mathcal{E}$, resulting in a lower throughput. We experimented with different bin-widths p (and hence different number of bins) and different values of a and observed that the overall throughput is always high when there is only a single bin (which implies that the scaling factor is 1) in the system (cf. Figure 7(c)).

- *Approximation Ratio:* We compare this approximation ratio obtained from the simulations with the theoretically estimated approximation ratio (ratio of the upper bound and lower bound). We observe that the upper bound as determined by the LP monotonically increases with the number of connections and has a value of $(k \cdot cap)$, where k denotes the number of connections and $cap = 5Mbps$ is the bandwidth of the system (cf. Figure 7(b)). The lower bound on the other hand is 1 as we use a single bin in the simulations. We observe that the approximation ratio is much lower than the estimated theoretical approximation ratio. This shows that in practice our approximation techniques provide a better approximation ratio than predicted and the bounds derived by our methods are indeed worst-case bounds. This result indicates that the derived upper and lower bounds are weak and additional research is required to improve these bounds.

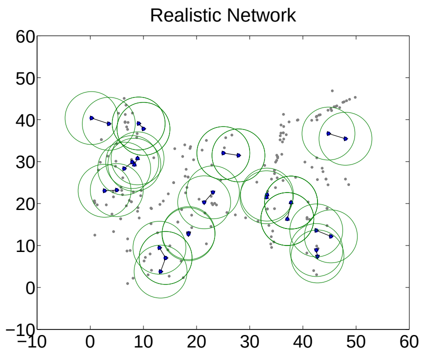
- *Comparison with 802.11* We observe that the results of our approximation techniques are comparable with the 802.11 protocol (cf. Figure 7(b)). 802.11 is a distributed random-access protocol and its performance is very close to our centralized technique. However, it should be noted that in order to ensure a fair comparison, the 802.11 simulations were conducted with a fairly high queue size of 12.5MB (≈ 8000 packets). For 802.11 simulation, the rate at which packets arrive at every transmitter was set to the bandwidth (5Mbps in this case); this ensured that the MAC layer always has a packet to send. In order to prevent packet loss due to saturated queues, the queue sizes for all transmitters were set to a high value. We observed that the 802.11 throughput decreased significantly when the queue sizes were set to the most commonly used value of 50KB [18] (cf. 7(d)). In Figure 8 we demonstrate the increase in the queue size at a particular node with simulation time.

We conclude that (a) the bounds derived by our techniques are indeed worst case approximation bounds and in practice our methods perform better than predicted. Additional research is however required to improve the upper and lower bounds, (b) the performance of the approximation algorithm is influenced by the length-diversity, (c) higher performance gains can be achieved by engineering the system. Partitioning the edges into different bins provided less throughput gains than using a single bin and, (d) the congestion measure proposed in this work can lead to overly pessimistic estimates for high traffic regimes. Additional research is required in developing an efficient congestion measure.

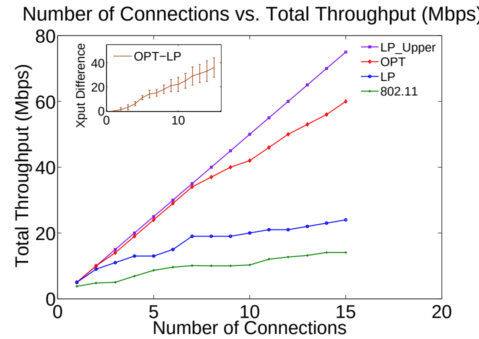
10 Improved Approximations for Grid Topologies

We now consider a special case of topology called the grid topology, where in the nodes are placed on a uniform spacing grid. The approximation obtained from Theorem 1 holds for any arbitrary graph topologies. We show that the poly-log approximation derived in Section 6 can be improved to a constant factor approximation for the case of grid topologies.

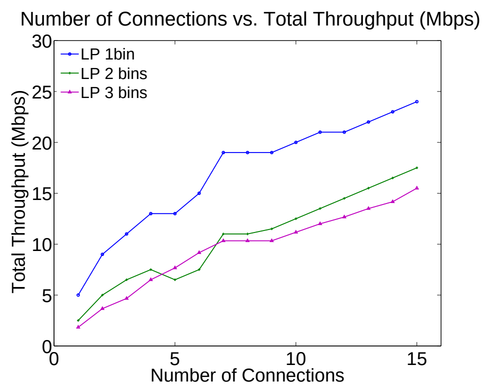
For a set V of n nodes, consider a grid of $\sqrt{n} \times \sqrt{n}$ with uniform grid spacing of d units. Let the nodes be placed on the grid points (cf. Figure 9). Let $E \subseteq V \times V$. For the grid topology, we consider an input instance of TM-SINR specified as $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$. We use the problem formulation $\mathcal{P}(\lambda, \mathcal{I})$ as discussed in Section 6.



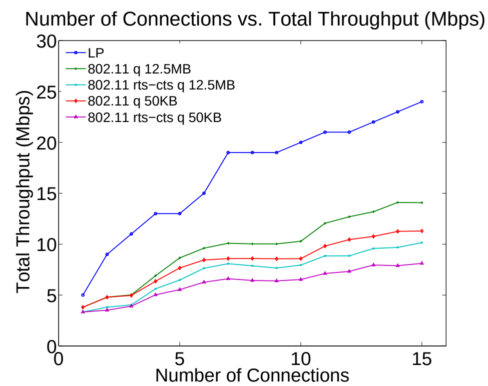
(a) Instance of the realistic network for a particular seed. One-hop source-destination pairs are selected u.a.r.



(b) Variation of throughput w.r.t. the number of connections as observed by the optimal solution, LP and 802.11 protocol. LP_Upper denotes the upper bound computed by running the LP with the necessary conditions. The 802.11 protocol ran without rts-cts and the queue size was fixed to 12.5MB. The total number of bins in the LP was set to one.



(c) Different binning strategies were considered for the LP.



(d) 802.11 protocol was considered with and without rts-cts messaging, for different queue sizes, 12.5Mbytes and 50Kbytes respectively.

Figure 7: Simulation results comparing LP with the optimal solution and 802.11.

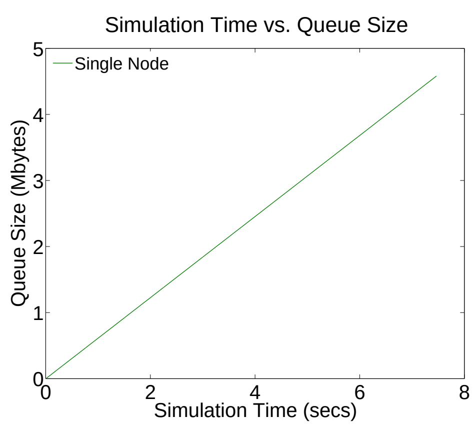


Figure 8: Variation in queue size w.r.t simulation time. For the 802.11 protocol, queue size is observed at a particular node.

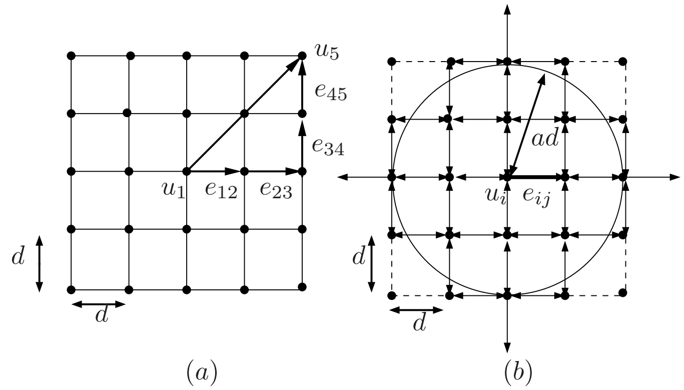


Figure 9: (a) Demonstrating Manhattan routing: For a grid with uniform grid spacing of d, the flow on direct edge $e_{15} = (u_1, u_5)$ can be replaced by flows on short edges $e_{12}, e_{23}, e_{34}, e_{45}$. (b) Demonstrating maximum number of interfering links for edge e_{ij}: Following the definition from C(e), all the solid edges interfere with edge e_{ij}

Observation 1 *[6] For a $\sqrt{n} \times \sqrt{n}$ grid, with uniform grid spacing of d units, for an instance $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ of* TM-SINR*, a flow $f(e_{ij})$ obtained by solving $\mathcal{P}(\lambda, \mathcal{I})$ on any direct edge $e_{ij} = (u_i, u_j)$ between nodes $u_i, u_j \in V$ where $\ell(e_{i,j}) > d$, can be replaced by flows on short edges of length d using Manhattan routing.*

Proof For any node u_i, let $row(i)$ denote the set of nodes that have the same $y - coordinate$ as that of u_i and let $col(i)$ denote the set of nodes that have the same $x - coordinate$ as that of u_i. For a given pair of nodes, u_i, u_j, a node u_k appears in the Manhattan routing path from u_i to u_j if and only if, $u_i \in row(u_k)$ or $u_j \in col(u_k)$ and both u_i, u_j lie at a distance less than $\ell(e_{ij})$ from u_k. We select all nodes in the Manhattan routing path, such that distance between any two adjacent nodes u_k, u'_k is equal to d. Therefore all the edges in the Manhattan routing path have length equal to d. The flow of $f(e_{ij})$ on edge e_{ij} can now be replaced by flows on edges on the Manhattan routing path (cf. Figure 9). It can be seen that this Manhattan routing scheme satisfies flow conservation constraints. ■

We now show that by using the Manhattan routing scheme, the total throughput achieved for a grid topology is within a constant factor away from the optimal solution.

Lemma 10 *Let $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ be an instance of the* TM-SINR *problem for a $\sqrt{n} \times \sqrt{n}$ grid, with uniform grid spacing of d units, and let $\bar{x} \in \mathcal{X}(\mathcal{I})$ be any feasible* link utilization *vector obtained using Manhattan routing. Then, the necessary conditions for link flow schedulability are:*

$$\forall e \in E, \forall k \in M, \sum_{e' \in C(e) \cap H^k} x(e') \leq \lambda.$$

Proof The proof is similar to that of Lemma 3. Since we are using the Manhattan routing, all the schedulable edges have the same length d. Partitioning based on edge lengths is therefore not required. We only need to consider sets $H_k = \{j_{min} \cdot 2^k \leq J(e) < j_{min} \cdot 2^{k+1}\}, \forall k \in M, \forall e \in E$. The Lemma therefore follows by making this minor modification to the proof of Lemma 3. Note that the above necessary conditions can be extended for the case of generic, uniform and *linear* power levels by replacing λ with λ_0, λ_1 and λ_2 respectively. ■

Lemma 11 *For a $\sqrt{n} \times \sqrt{n}$ grid, with uniform grid spacing of d, with Manhattan routing, the sufficient conditions for link flow schedulability are as follows:*

$$\forall e \in E, \forall k \in M, \sum_{e' \in C(e) \cap H_k} x(e') \leq \frac{\lambda}{c'},$$

where $c' = (2(2a)(2a+1)) - 4$, *a is a constant defined in Section 4.3 and is an integral multiple of* d.

Proof For any transmitting edge $e_{ij} = (u_i, u_j) \in E$ consider a disk D of radius $a \cdot d$ centered at node u_i. Disk D is contained within a grid of dimension $2a \times 2a$ (cf. Figure 9). Since we are using Manhattan routing, all the schedulable edges have length equals d. From the definition of $C(e_{ij})$ (cf. Section 4.3) we know that all the edges $e' = (u', v')$ that have their transmitters u' within disk D interfere with edge e_{ij}. The total number of directed edges of length d in a $2a \times 2a$ grid are $8a(2a+1)$. By excluding the corner edges, the maximum number of edges that can interfere with a given edge e_{ij} is a constant and is obtained as $c' = 8a(2a+1) - 12$. We now scale the *link utilization* by a factor c' to obtain the following sufficient conditions,

$$\forall e \in E, \forall k \in M, \sum_{f \in C(e) \cap H_k} x(e) \leq \frac{\lambda}{c'}.$$

■

Theorem 4 *Let* $\mathcal{I} = (V, E, \mathcal{D}, \bar{J})$ *be an instance of* TM-SINR *for a* $\sqrt{n} \times \sqrt{n}$ *grid, with uniform grid spacing of* d. *The optimum solution* $\bar{x}$ *to the program* $\mathcal{P}(\lambda/c', \mathcal{I})$ *is a feasible and stable* link utilization *vector for the instance* $\mathcal{I}$, *and results in a total throughput of at least* $\Omega(r_{opt}(\mathcal{I})/c')$.

Proof Let $\bar{x}_{opt}$ be the optimum utilization vector for the instance $\mathcal{I}$ of TM-SINR, achieving a total throughput rate of $r_{opt}(\mathcal{I})$. From Lemma 10, it follows that $\bar{x}_{opt}$ is a feasible solution to the program $\mathcal{P}(\lambda, \mathcal{I})$. Further, $\bar{x}_{opt}/c'$ is a feasible solution to the program $\mathcal{P}(\lambda/c', \mathcal{I})$ and results in a total throughput rate of $\frac{r_{opt}(\mathcal{I})}{c'}$. Therefore, the theorem follows. ■

11 Conclusion

We study the problem of throughput maximization in arbitrary wireless networks with SINR constraints from a theoretical perspective, and take the first steps toward developing efficient algorithms for this problem. Our results show that the comparison between SINR and graph-based models is complicated, and for different instances, different models might give higher estimates of the throughput capacity, suggesting the need for greater care in using these models. We develop the first provable algorithms for approximating the throughput capacity in the SINR models by means of a linear programming formulation, extending the recent work of [7, 17].

Extending these results to distributed algorithms would make them more useful from a practical point of view. This paper does not consider power control, and studying the problem of joint power control and throughput maximization would be an interesting extension. We only consider the AWGN model for specifying the link capacities, extending this model to include the SINR [1] would also be an interesting problem.

Acknowledgments

We thank the referees for their helpful suggestions.

References

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

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

[3] Baruch Awerbuch and Rohit Khandekar. Greedy distributed optimization of multi-commodity flows, *Distributed Computing*, vol 21, pp.317-329, 2009.

[4] N. Bansal and Z. Liu. Capacity, delay and mobility in wireless ad-hoc networks. In *Proc. IEEE INFOCOM*, pages 58–72, April 2003.

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

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

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

[8] D. Chafekar, V.S. Anil Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Approximation Algorithms for Computing the Capacity of Wireless Networks with SINR constraints. In *Proc. IEEE INFOCOM*, pages 1166-1174, April 2008.

[9] Alexander Fanghnel, Thomas Kesselheim, and Berthold Vcking. Improved algorithms for latency minimization in wireless networks. *Theoretical Computer Science (conference version in ICALP 2009)*, In Press, Corrected Proof:–, 2010.

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

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

[12] B. Han, V.S. Anil Kumar, M. Marathe, S. Parthasarathy and A. Srinivasan. Distributed Strategies for Channel Allocation and Scheduling in Software-Defined Radio Networks, *IEEE INFOCOM*, pp. 1521-1529, 2009.

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

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

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

[16] U. Kozat and L. Tassiulas. Throughput capacity in random ad-hoc networks with infrastructure support. In *Proc. ACM MOBICOM*, pages 73–87, September 2003.

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

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

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

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

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

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

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

[24] TRANSIMS Open Source Project. http://http://www.transims-opensource.net/.

[25] R. Ramanathan. A unified framework and algorithm for (t/f/c) dma channel assignment in wireless networks. In *Proc. IEEE INFOCOM*, pages 900–907, April 1997.

[26] G. Sharma, R. Mazumdar, and N. Shroff. On the complexity of scheduling in wireless networks. In *Proc. ACM MOBICOM*, pages 227–238, September 2006.

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

[28] S. Toumpis and A. Goldsmith. Capacity regions for wireless ad hoc networks. In *Proc. International Symposium on Communication Theory and Applications*, pages 227–238, April 2001.