Cross-Layer Latency Minimization in Wireless Networks with SINR Constraints

Deepti Chafekar
Department of Computer
Science
Virginia Tech
Blacksburg, VA 24061
chafekar@vt.edu

V.S. Anil Kumar
Department of Computer
Science and Virginia
Bioinformatics Institute
Virginia Tech
Blacksburg, VA 24061
akumar@vbi.vt.edu

Madhav V. Marathe
Department of Computer
Science and Virginia
Bioinformatics Institute
Virginia Tech
Blacksburg, VA 24061
mmarathe@vbi.vt.edu

Srinivasan Parthasarathy
Next Generation Distributed
Systems
IBM T. J. Watson Research
Center
Yorktown Heights, NY 10598
spartha@us.ibm.com

Aravind Srinivasan
Department of Computer
Science and Institute for
Advanced Computer Studies
University of Maryland
College Park, MD 20742
srin@cs.umd.edu

ABSTRACT

Recently, there has been substantial interest in the design of cross-layer protocols for wireless networks. These protocols optimize certain performance metric(s) of interest (e.g. latency, energy, rate) by jointly optimizing the performance of multiple layers of the protocol stack. Algorithm designers often use geometric-graph-theoretic models for radio interference to design such cross-layer protocols. In this paper we study the problem of designing cross-layer protocols for multi-hop wireless networks using a more realistic *Signal to Interference plus Noise Ratio (SINR)* model for radio interference. The following cross-layer latency minimization problem is studied: Given a set V of transceivers, and a set of source-destination pairs, (i) choose power levels for all the transceivers, (ii) choose routes for all connections, and (iii) construct an end-to-end schedule such that the SINR constraints are satisfied at each time step so as to minimize the make-span of the schedule (the time by which all packets have reached their respective destinations). We present a *polynomial-time algorithm with provable worst-case performance guarantee* for this cross-layer latency minimization problem. As corollaries of the algorithmic technique we show that a number of variants of the cross-layer latency minimization problem can also be approximated efficiently in polynomial time. Our work extends the results of Kumar *et al.* (*Proc. SODA*, 2004) and Moscibroda *et al.* (*Proc. MOBIHOC*, 2006). Although our algorithm considers multiple layers of the protocol stack, it can naturally be viewed as compositions of tasks specific to each layer — this allows us to improve the overall performance while preserving the modularity of the layered structure.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*; F.2.3 [**Theory of Computation**]: ANALYSIS OF ALGORITHMS AND PROBLEM COMPLEXITY—*Trade-offs between Complexity Measures*

General Terms

Algorithms, Theory

Keywords

Cross-layer design, interference, end-to-end scheduling, wireless networks, SINR model

1. INTRODUCTION

Recently, there has been substantial interest in the design of cross-layer protocols for wireless multi-hop networks: see [6, 10, 19] and the references therein. The work is motivated by two primary observations: (i) the performance of current protocols is far from satisfactory, and classical paradigms based on wireline networks are unlikely to scale; and (ii) wireless multi-hop networks are being used in diverse application domains that often force designers to optimize different performance-objectives.

This paper considers the problem of minimizing the make-span of schedules in wireless multi-hop networks by *jointly* considering the routing, scheduling and power control layers. Recall that the make-span (or length) of a schedule is the time by which the last packet has reached its destination; thus, we have a min-max (latency) metric. Our goal is to develop polynomial-time algorithms with provable worst-case performance guarantees. A more general form of the problem we study is informally stated as follows: Given (i) a set of points V in the plane at which transceivers are located, (ii) gain β and path-loss exponent α, (iii) source-destination pairs $K = \{(s_1, t_1), \ldots, (s_k, t_k)\}$, (iv) a set of packets Q for all k connections, and (v) a power range $[p_{min}, p_{max}]$, the *Cross-layer latency minimization* (CLM) problem is to (a) assign power levels to individual transceivers, (b) choose routes for the packets, and (c) construct an end-to-end schedule for the packets such that all

the transmissions that happen simultaneously satisfy the MAC interference constraints, so that the overall latency is minimized; the *overall latency* or *make-span* is the time elapsed before all packets successfully reach their respective destinations. Variations of the problem where the solution to one or more layers is already provided (e.g., routes are already given) are discussed in Section 1.1.

In order to fully formalize the CLM problem, interference at the MAC layer needs to be modeled. Since wireless interference is quite complex, discrete combinatorial and geometric models are commonly used: the wireless network is modeled as a graph $G = (V, E)$, in which link $(u, v) \in E$ implies that node v can receive the signal from node u. A disk is associated with each node to specify its range. Interference in such a graph is modeled through independence constraints (see e.g., [26]): *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 [3,15,26,27]. Clearly, this abstraction is an oversimplification of the real process – the signal from a radio does not end abruptly at a boundary, and signal collision does not always lead to lost messages. In reality, 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; this is commonly known as the *Signal to Interference Plus Noise Ratio (SINR)* model [10, 21, 22]. The difficulty of analyzing algorithms using SINR models and obtaining potential performance gains in theory as well as in practice has been discussed in [21, 22], and is due to the following two reasons. First, SINR based models usually lead to non-convex constraints. Further, by including SINR constraints, problems such as routing, scheduling, power-control etc. sometimes become NP-hard; thereby rendering traditional convex-programming-based techniques inapplicable. Second, the geometric-graph-theoretic models *localize* the interference of a transceiver. This makes the analysis tractable; however, the independence constraints put severe and often artificial restrictions on scheduling algorithms in their ability to schedule links that are close to each other.

1.1 Overview of results and techniques

The CLM problem is NP-hard to solve exactly. The proof follows by restriction: the minimum-length link-scheduling problem for the SINR model is NP-hard [1]. Since this problem is a special case of the CLM problem in which all the connections are on adjacent links, it follows that the CLM problem with SINR constraints is also NP-hard. Therefore, we focus on approximation algorithms. Our goal is to devise polynomial-time approximation algorithms with provable worst-case performance bounds [29]. Specifically, we obtain a bi-criteria approximation algorithm: given a range $[p_{min}, p_{max}]$ from which the power level of each node must be chosen, as well as some parameter $\epsilon > 0$, we present an efficient algorithm which results in the overall latency being at most a polylogarithmic factor more than that of an optimal solution for the problem in which all power-levels are constrained to lie in $[p_{min}, (1 - \epsilon)p_{max}]$: see Theorem 2. Logarithms such as those of Theorem 2 are to the base two, unless specified otherwise. Also, the terms such as "$\log \Delta$" and "$\log \Gamma$" should be interpreted as "$1 + \log \Delta$" and "$1 + \log \Gamma$" to ensure that such quantities are bounded away from 0; we do not do this explicitly, in order to avoid notational clutter.

Here we describe a general algorithmic approach and use this approach to design a polynomial-time randomized approximation algorithm (MinDelay) for the CLM problem, with rigorous provable bounds on the overall latency compared to the optimal solution. Note that this is a worst-case approximation guarantee, i.e.,

this holds for every instance, and for special instances (e.g., random), our algorithms might have much better performance guarantees. In addition, we do not need to know the optimal solution in order to prove the approximation guarantees on our algorithm. To our knowledge, this is the first result to extend the link scheduling work of Moscibroda et al. [21] to the CLM problem, with SINR constraints. While our algorithm solves the CLM problem, it can be naturally decomposed into specific tasks at each layer of the protocol stack combined with appropriate information-exchange between these layers. This allows us to improve the overall performance and simultaneously preserve the modularity provided by the layered architecture. As corollaries of the algorithmic and lower bounding techniques, we obtain polynomial-time randomized approximation algorithms for the following additional problems: (a) joint routing and end-to-end scheduling to minimize latency, when the *power levels are pre-specified* for all nodes; (b) end-to-end scheduling to minimize latency, when both the routes and power levels are pre-specified; and (c) the CLM problem with an additional constraint that the total energy consumed is at most some given B: our algorithm can be modified so that the latency is optimized, with the total energy used being at most $(B \log n)$, where n is the number of transceivers. It is important to note that in general, being able to solve the CLM problem efficiently *does not* immediately mean that one can solve these variants efficiently.

Our algorithm MinDelay combines two powerful techniques in order to solve the CLM problem: linear-programming rounding for path selection, and random delays for the end-to-end scheduling. In addition, we have introduced a number of new technical ideas. First, we introduce a new congestion measure that incorporates the SINR constraints, and show that the overall latency can be related to this congestion measure. Our formulation of the linear program (LP) and the lower bounds crucially depends on the new congestion measure. In contrast to the congestion measures proposed in [14, 16], the new congestion measure needs to account for the SINR model and is thus not purely graph-theoretic. It is easy to construct examples where using the congestion measure of [16] would give much higher latency. Second, while the random-delays technique is well-known, it requires much more involved analysis in our case in order to derive good bounds. Finally, to prove our upper bounds, we extend a very useful technique given in [22].

We believe that the algorithmic techniques and models described in this paper can increase overall network-performance. As mentioned earlier, our algorithms use the SINR model of interference. It has been shown in [20] that SINR models can obtain significant performance gain in theory and practice. Further, [22] demonstrates that using SINR models for link-layer scheduling can increase overall performance and decrease link-delays. The problem studied in this paper can be viewed as a special case of the problem discussed in [22]. Further, we use congestion control for routing and scheduling. Congestion-aware routing and scheduling has been shown to improve performance gains in [4, 5].

We note that the algorithms presented in this paper are theoretical in nature. Nevertheless, our theoretical results suggest an interesting class of congestion-aware link-metrics that can be used to guide the design and performance analysis of realistic protocols. Furthermore, the mathematical analysis and the congestion measure proposed here are likely to be of independent interest and applicable to a more wide class of problems. For example, we are currently exploring similar algorithms for throughput maximization and design of congestion-aware routing protocols.

Citation	Layers	Objective	Interference model	Type of algorithm
[2]	R, S, P	Min-P	link	centralized, approximation
[8]	R, S, P	Min-P	SINR	centralized
[17]	Ra, S	Max-Rate	link	centralized
[18]	Ra, S	Max-Rate	SINR,link	distributed (for restricted interference model)
[4]	R, S	Max resource utilization	SINR,link	centralized and distributed(for scheduling)
[12]	Ra, R, S	Max-Rate	link	centralized
[9]	S, P	Min-P	SINR,link	(only for power control), distributed heuristic
[13]	S, P, Ra,	Max-Rate	SINR	centralized, heuristic
[21]	S, P	Min-D (not end-to-end)	SINR	centralized
[14]	S, R	Min-D (end-to-end)	Disk	centralized and distributed

Table 1: Some papers related to this work. R denotes Routing, S denotes Scheduling, P denotes Power control and Ra denotes Rate control. Min-P denotes minimizing power, Min-D denotes minimizing delay, and Max-Rate denotes maximizing data rates

2. RELATED WORK

Cross-layer optimization has an extremely active area of research in recent years [2,4–6,8,9,12,13,17–19,22,23]. In Table 1, we compare our work with some of the most relevant previous results. Recent work by [2] presents a polynomial-time 3-approximation algorithm for joint routing, scheduling and power control problem. The authors however use a weaker interference model of synchronous time-slotted TDMA/CDMA. Similar work by [8] presents an integrated routing, link scheduling and power-allocation policy with the SINR model of interference. The authors present an algorithm for minimizing total average power of a multi-hop wireless network by considering link scheduling and power control. The algorithm presented appears to be efficient, however as acknowledged in [8] the algorithm suffers from a worst-case exponential complexity as a function of the number of transmission nodes. Recent work by [4, 5] considers the problem of congestion control and resource allocation (via routing and scheduling). The paper aims to maximize resource allocation by using a duality approach to decompose the main problem into sub-problems of congestion control and routing/scheduling. The works [12, 17, 18] study the joint problem of allocating data rates and finding a stabilizing scheduling policy in a multi-hop wireless network. However, our problem is different than the ones considered in [4,5,12,17,18]; we consider the problem of minimizing end-to-end delay by jointly optimizing routing, scheduling and power control.

Our work is most closely related to [21, 22] and [14]. The paper [21] studies the problem of scheduling packets with SINR constraints to ensure strong-connectedness of the graph. This is a topology-control problem that aims at finding a schedule of minimal length in which all transmitted links form a strongly-connected network. In [22], the authors study the MAC level scheduling problem with SINR constraints. The authors present a scheduling algorithm that successfully schedules a set of links in polylogarithmic time by assigning non-uniform power to the transmitting nodes. They do not compare the performance of their algorithm with the optimal schedule. Further, the non-linear power assignment is exponential in the number of nodes. Therefore for large networks, the power assignment for individual nodes could be very high, thereby increasing the overall power-consumption of the network. There are crucial differences between the work presented here and those of [14, 21, 22]. First, the results of [22] do not directly give bounds on the end-to-end latency – their result is mainly applicable for scheduling a given set of edges, and does not suggest which set of waiting packets to schedule first. The second important difference is in the power control step: the specific power-level choice

in [22] helps to minimize the link-layer delays. During the end-to-end scheduling problem, the same node v might have to transmit along different links (v, w) and (v, w') at different times. Using the algorithm from [22] directly might result in different power assignments for v at different times. In contrast, we focus on choosing a single power-level for each node, so that the overall latency is minimized. The paper [21] studies the scheduling complexity of connecting a given number of nodes located at arbitrary positions by some communication tree. Our problem is slightly different from the one of [21]: for given source-destination pairs, we initially find paths, and then schedule packets on the selected paths. Finally, although the work in [14] does deal with end-to-end latencies, the algorithm and its analysis is based on geometric-graph models of radio-interference.

3. PRELIMINARIES AND MODELS

Let V denote the set of transceivers located in the plane – we will sometimes refer to these transceivers as nodes. Let $d(u, v)$ denote the Euclidean distance between any two nodes $u, v \in V$. Sometimes we will consider a link $e = (u, v)$, and we will use $\ell(e) = d(u, v)$ to denote its length. We let

$$\Delta = \max_{u,v \in V} d(u, v) / \min_{u,v \in V: \, u \neq v} d(u, v)$$

be the ratio between the maximum and minimum inter-point separation. Without loss of generality, we assume that the minimum distance between any pair of nodes is 1. If $J(u)$ denotes the power assigned to node u, we denote

$$\Gamma = \frac{\max_u J(u)}{\min_{u'} J(u')}.$$

3.1 The SINR Physical Model

We use the SINR model of interference as described in [21]. In this setting, for any given link $e = (u, v)$ with u as the sender and v as the receiver, the received signal power $J_v(u)$ at node v due to sender u, can be expressed as $J_v(u) = \dfrac{J(u)}{d(u, v)^\alpha}$ where $J(u)$ denotes the transmission power of the sender u and α denotes the path-loss exponent. As suggested in [22] we assume $\alpha > 2$. Table 2 gives a list of most of the notation used in this paper. For a given set $E' = (u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)$ of simultaneously-communicating links, the interference $I_r(v, E')$ at receiver v from

all the senders is

$$I_r(v, E') = \sum_{e'=(u',v')\in E', u'\neq u} \frac{J(u')}{(d(u',v))^\alpha}.$$

Whenever the set E' is clear from the context, we will simply denote this as $I_r(v)$. The node v can successfully receive transmission from u, if

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

where N denotes the ambient noise (some given constant) and β denotes the antenna gain.

3.2 Congestion and Dilation

Our algorithms and their analyses are based on two important quantities: *congestion and dilation*. Let P_i be the path chosen for connection (s_i, t_i), and let $\mathcal{P}$ be the collection of these paths. Let $E(P_i)$ denote the set of edges in P_i. We define a multi-set $E = E(\mathcal{P}) = \cup_i E(P_i)$. We define the dilation $D(\mathcal{P})$ as the length of the longest path in $\mathcal{P}$. We next define the congestion C induced by $\mathcal{P}$ through a definition of "interference sets" $C(e)$:

DEFINITION 1.

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

Note that a is a constant. In our analysis, we would require

$$a \geq \sqrt[\alpha]{2^\alpha \frac{96\beta(1+\epsilon)}{\epsilon(\alpha-2)}} \ \text{and} \ \alpha > 2.$$

and $a \geq 2$. Here ϵ is a small positive constant. It can be seen from Lemma 2 that these lower bounds on a and α are required for the validity of the final schedule. Figure 1 demonstrates the 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. Our definition of congestion is motivated by the definition of congestion used in [14, 16] but has differences in order to account for the SINR model of radio interference.

4. THE LATENCY-MINIMIZATION ALGORITHM

We now present the paper's main technical contribution, the algorithm MinDelay (Algorithm 1) for joint power control, end-to-end scheduling and routing to minimize the overall latency. In Section 5 we show that for any input instance, the latency achieved by algorithm MinDelay is within a polylogarithmic factor of the optimal latency for that instance. We first describe the algorithm informally. It has three parts: PathSelection selects paths for each connection, PowerControl assigns a power level to each sender that lies on the selected paths, and Schedule schedules the packets on their respective paths. Although the three procedures are modular, there is significant interplay between these three components and they jointly minimize the end-to-end latency. These procedures are described in the following subsections.
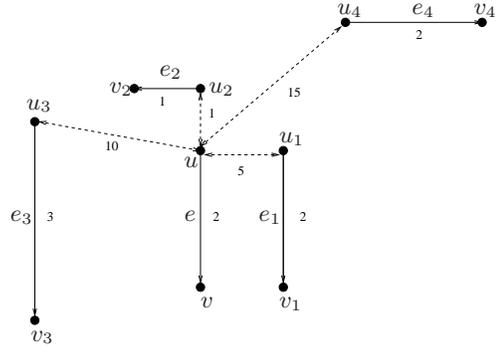

Figure 1: Illustrating congestion measure for a link. The solid lines represent edges along with edge lengths (e.g., $l(e) = 2$). The dotted lines represent distance between two nodes (e.g., $\mathbf{d(u, u_1) = 5}$). Let $\mathbf{a = 6}$. Then $\mathbf{C(e) = \{e, e_1, e_3\}}$ for link $\mathbf{e = (u, v)}$. By definition, $\mathbf{e_4, e_2 \notin C(e)}$ since $\mathbf{a \cdot d(u_4, v_4) < d(u, u_4)}$ and $\mathbf{d(u_2, v_2) < d(u, v)}$.

Algorithm 1: MinDelay

Input : $V, \mathcal{E}, K, [p_{min}, p_{max}], Q, \alpha, \beta$
Output: (i) $\mathcal{P}$ (ii) J, (iii) A valid schedule $\mathcal{S}$ that moves packets along their paths to the destinations successfully

1 $\mathcal{P} = \mathsf{PathSelection}(V, K, \mathcal{E}, [p_{min}, p_{max}])$, $E = E(\mathcal{P})$
2 $J = \mathsf{PowerControl}(V, K, \mathcal{P}, E)$
3 $\mathcal{S} = \mathsf{Schedule}(V, K, \mathcal{P}, Q, J, E)$

Procedure PathSelection

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

//Solve the LP
1 Solve the LP as described in equations 1..7 to get the solutions $(\{y_f(i,e)\}, w_f)$.
//Path Decomposition
2 Decompose the flow into path flows — let $\mathcal{M}_i$ denote the set of paths from s_i to t_i, with $h(P)$ denoting the flow on path $P \in \mathcal{M}_i$. Also let $|P|$ denote the length (number of links) of path P.
//Path Filtering
3 **foreach** $P \in \mathcal{M}_i$ **do**
4 $\quad \gamma = \sum_{P' \in \mathcal{M}_i : |P'| > 2w} h(P')$.
5 $\quad$ **if** $|P| \leq 2w$ **then**
6 $\quad\quad z(P) = h(P)/(1 - \gamma)$.
7 $\quad$ **else**
8 $\quad\quad z(P) = 0$
9 $\quad$ **end**
10 **end**
//Randomized Rounding
11 **foreach** i **do**
12 $\quad$ choose exactly one path $P \in \mathcal{M}_i$ with probability $z(P)$.
13 $\quad$ Let $P_i = P$ be the path chosen from $\mathcal{M}_i$
14 **end**
15 Let $\mathcal{P} = \{P_1, \ldots, P_k\}$.

Procedure PowerControl

Input : $V, K, \mathcal{P}$, multi-set $E = E(\mathcal{P})$
Output: J
1 Choose power levels $J(u)$ for each node u as
$J(u) = \max_{e=(u,v)\in E}\{(1+\epsilon)\beta N d(u,v)^\alpha\}$

Procedure Schedule

Input : $V, K, Q, \mathcal{P}, E, J$
Output: A valid schedule $\mathcal{S}$ that moves packets along their paths to their destinations

```
//Random Delay
```
1 **foreach** *packet q_i at source s_i* **do**
2   Choose a delay $X_i \in \{1, .., C\}$ uniformly at random
3 **end**
```
//Construct an invalid schedule S'
```
4 **foreach** *packet q_i* **do**
5   wait at source s_i for time X_i before starting
6   move one edge at a time along P_i after waiting X_i steps, i.e., move on the jth edge of P_i at step $X_i + j$
7 **end**
```
//Convert S' into a valid schedule S
//|S'| denotes the length (make-span) of S'
```
8 **for** $t = 1, 2, \ldots, |\mathcal{S}'|$ **do**
```
     //Construct a partial schedule S_t
```
9   E_t = set of links scheduled by $\mathcal{S}'$ in time t.
10      $\mathcal{C}_t(e) = |C(e) \cap E_t|, \forall e \in E_t$
```
      //Set partitioning
```
11   Partition E_t into sets $B = \{B_0, B_1, \ldots, B_{\lfloor \log \Delta \rfloor}\}$ such that $B_i = \{e \in E_t : 2^i \leq \ell(e) < 2^{i+1}\}$
12   **for** $i = \lfloor \log \Delta \rfloor$ *down to* 0 **do**
13   Partition links in set B_i into sets $W_i = \{W_0^i, W_1^i, ..., W_{\lfloor \log \Gamma \rfloor}^i\}$ such that $W_j^i = \{e = (u, v) \in B_i : 2^j \leq J(u) < 2^{j+1}\}$
14   **for** $j = \lfloor \log \Gamma \rfloor$ *down to* 0 **do**
```
               //Greedy Coloring
```
15   Order the edges in $E_t \cap W_j^i$ in non-increasing order of length $\{e_1, e_2, \ldots, e_s\}$
16   **for** $k = 1$ *to* $|E_t \cap W_j^i|$ **do**
17   Let $f(e_k)$ be a positive integer denoting the color that is to be assigned to edge e_k
18   Assign e_k the smallest numbered color that is not used by any edge in $C(e_k) \cap \{e_1, \ldots, e_{k-1}\}$,i.e.,$f(e_k)$ is the smallest positive integer not in the set $\{f(e_{k'}) : k' < k, \ e_{k'} \in C(e_k)\}$.
19   **end**
20   Construct a schedule $\mathcal{S}_{t,i,j}$ for edges in $E_t \cap W_i^j$ by scheduling all edges e with color $f(e) = r$ in the rth step, for $r = 1, 2, \ldots$.
21   **end**
22   Construct schedule $\mathcal{S}_{t,i} = \mathcal{S}_{t,i,\log \Gamma} \bullet \ldots \bullet \mathcal{S}_{t,i,0}$ (*The $\bullet$ operator implies concatenation of different schedules*)
23   **end**
24   Construct schedule $\mathcal{S}_t = \mathcal{S}_{t,\log \Delta} \bullet \ldots \bullet \mathcal{S}_{t,0}$
25 **end**
26 Construct schedule $\mathcal{S} = \mathcal{S}_1 \bullet \ldots \bullet \mathcal{S}_\ell$, where $\ell = |\mathcal{S}'|$

n	number of nodes
V	set of nodes
$\mathcal{E} = \{(u, v) \in V \times V\}$	set of all edges
$K = \{(s_1, t_1), \ldots, (s_k, t_k)\}$	k source-destination pairs
$[p_{min}, p_{max}]$	power range
$Q = \{Q_1, \ldots, Q_k\}$	set of packets for k sources
$\mathcal{P} = \{P_1, \ldots, P_k\}$	set of paths for k connections
$E = E(\mathcal{P})$	set of edges on paths $\mathcal{P}$
$C(e)$	congestion set for link e
$C = \max_{e \in E} |C(e)| = \text{Congestion}$	max. cardinality of any $C(e)$
$D = \text{Dilation}$	length of longest path in $\mathcal{P}$
$J = \{J(u_1), \ldots, J(u_n)\}$	set of power assignments
Γ	ratio of max. to min. power assignments
Δ	ratio of max. to min. node-distances
$\mathcal{S}$	a valid schedule
$N_{out}(u)$	set of outgoing edges from u
$N_{in}(u)$	set of incoming edges to u
α	path-loss exponent
β	antenna gain
N	ambient noise
a	positive constant
ϵ	small positive constant

Table 2: Notation used in this paper

4.1 Path Selection

Procedure PathSelection deals with selecting *good* paths, i.e., paths with a "low" value of $C + D$; recall the definitions from Section 3. We formulate this problem as an integer program (IP), following the methods of [28]. Let $\mathcal{E} = \{(u, v) \in V \times V\}$ be the set of all edges. Let $y(i, e)$ be a flow-indicator variable for each connection i and edge $e \in \mathcal{E}$: $y(i, e)$ is 1 if the path for connection i passes through e, and $y(i, e) = 0$ otherwise. Let $N_{out}(u) = \{(u, v) : v \in V\}$ and $N_{in}(u) = \{(v, u) : v \in V\}$ denote the set of outgoing and incoming edges, respectively, from u. Let $L = \{(u, v) \in V \times V : p_{max} < (1 + \epsilon)\beta N d(u, v)^\alpha\}$ denote the set of links (u, v) that are infeasible if we require $J(u) \geq (1 + \epsilon)\beta N d(u, v)^\alpha$. The following is the natural integer programming (IP) formulation.

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

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

$$\forall i, \forall v \neq s_i, t_i : \sum_{e \in N_{out}(v)} y(i, e) = \sum_{e \in N_{in}(v)} y(i, e) \quad (2)$$

$$\forall e \in \mathcal{E} : \sum_i \sum_{e' \in C(e)} y(i, e') \leq w \quad (3)$$

$$\forall i : \sum_e y(i, e) \leq w \quad (4)$$

$$\forall i, \forall e \in L : y(i, e) = 0 \quad (5)$$

$$\forall i, e : y(i, e) \in \{0, 1\} \quad (6)$$

In the IP formulation, constraints (1) ensure that exactly one outgoing edge is selected from each source. Constraints (2) capture flow conservation. Constraints (3) ensure that $C(e) \leq w$ for each edge e, and constraints (4) guarantee that each path selected has length at most w. Constraints (5) ensure that the chosen paths do not include edges in L. Let $(\{y_{int}(i, e)\}, w_{int})$ denote the integer program solution. Solving the IP is NP-hard, and so we use the LP-relaxation by relaxing constraints (6) to obtain constraints of the form

$$\forall i, e : y(i, e) \leq 1. \quad (7)$$

Let $(\{y_f(i,e)\}, w_f)$ denote the linear program solution. The LP solution gives flows between the connections, instead of paths. So the PathSelection procedure first decomposes this flow into paths $\mathcal{M}_i$, with flow $h(P)$ for each $P \in \mathcal{M}_i$. See [28] for details on flow decomposition. However, we still have $|\mathcal{M}_i| > 1$, i.e., we have a number of paths that carry the flow. The filtering step discards long paths in each $\mathcal{M}_i$, and the final step uses randomized rounding [25] to choose paths; in the next subsection, we show that the resulting paths have congestion plus dilation $O(w_f \log n)$.

4.1.1 Analysis of procedure PathSelection

We now derive an approximation on the congestion plus dilation of the paths selected by the above procedure.

LEMMA 1. *In procedure* PathSelection *after the path filtering step the fractional solution z obtained satisfies the following properties:*

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

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

3. *For any edge e, $\sum_{e' \in C(e)} \sum_{P:e' \in P} z(P) \le 2w$.*

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

PROOF. Recall that in the randomized rounding step, each path P is selected with a probability $z(P)$. In order to prove the above theorem, we first calculate the expected congestion on a link e due to set of paths chosen. We then apply the Chernoff bounds [7] to obtain an upper bound on the total congestion produced by the randomized-rounding procedure.

Let $y(P) \in \{0, 1\}$ be an indicator variable such that $y(P) = 1$ if path P is selected in the rounding procedure. It can be seen that $Pr[y(P) = 1] = z(P)$. Since $y(P)$ is an indicator variable $E[y(P)] = z(P)$. Let $\mathcal{P}(y) = \{P : y(P) = 1\}$, be the set of paths chosen in the rounding procedure. For any edge e, $cong(e, \mathcal{P}(y)) = \sum_{e' \in C(e)} \sum_{P:e' \in P} y(P)$, therefore,

$$E[cong(e, \mathcal{P}(y))] = \sum_{e' \in C(e)} \sum_{P:e' \in P} Pr[y(P) = 1]$$
$$= \sum_{e' \in C(e)} \sum_{P:e' \in P} z(P).$$

We know from Lemma 1 that

$$\sum_{e' \in C(e)} \sum_{P:e' \in P} z(P) \le 2w.$$

We have

$$E[cong(e, \mathcal{P})] \le 2w.$$

Since the variables $y(P)$ are independent, we can now apply the Chernoff bounds.

$$Pr[cong(e, \mathcal{P}(y)) \ge 3 \log n \max\{2w, 1\}] \le 2^{-3 \log n \max\{2w, 1\}}$$
$$\le \frac{1}{n^3}.$$

Since the number of edges is $O(n^2)$, by union bound we get

$$Pr[\exists\, e \text{ s.t. } cong(e, \mathcal{P}(y)) \ge 3 \log n \max\{2w, 1\}] \le \frac{1}{n}.$$

Therefore with high probability, the congestion of the solution produced by the randomized rounding procedure is at most $3 \log n \max\{2w, 1\}$. Since $w \le w_{int}$, the congestion produced by the randomized rounding procedure is $O(w_{int} \log n)$, which is an $O(\log n)$-approximation. $\square$

We note that the PathSelection step can be derandomized using standard techniques from [24].

4.2 End-to-end Scheduling

At this point, we have already chosen a path P_i for each connection (s_i, t_i), and the packets need to be scheduled along these paths so that the overall latency is minimized. Procedure Schedule is based on the 'random delays' technique of Leighton, Maggs and Rao [16], and on the techniques used to adapt this to geometric graphs in [14]. The congestion and dilation are important because we show that they are good lower bounds on the optimum, and it is possible to construct a schedule with length proportional to the congestion+dilation. The following is an informal description of the main steps in Procedure Schedule; see the procedure-description for the details.

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

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

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

4. Greedy Coloring: After set partitioning, we color the links in each W_j^i using a greedy coloring scheme.

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

4.2.1 Analysis of procedure Schedule

The analysis of procedure Schedule consists of the following parts: (i) Validation, where we prove that the schedule produced by Schedule is indeed valid, i.e., that the SINR constraints are satisfied at each step, (ii) Length of the schedule, where we derive an upper bound on the length of the schedule produced by Schedule and (iii) Length of the optimal schedule, where we derive a lower bound on the length of any optimal schedule in which the maximum power assigned to any node is constrained to be at most $(1 - \epsilon)p_{max}$.

Validity of Schedule

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

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

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

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

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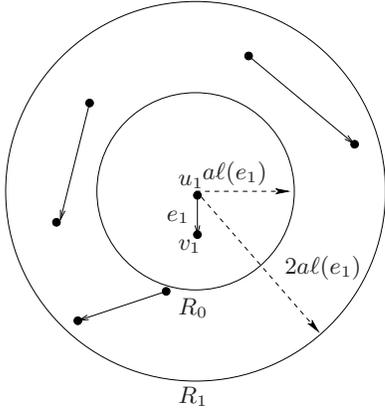

Figure 2: For a given link $\mathbf{e_1 = (u_1, v_1)}$, construct rings of radius $\mathbf{a\ell(e_1)}$ around $\mathbf{u_1}$. We calculate the interference experienced by node $\mathbf{v_1}$ due to other simultaneously transmitting links.

We will now calculate the $SINR(v_1)$ due to all the other transmissions in $\mathcal{T}$. As in [22], we will partition the plane into rings centered at u_1, in order to compute the interference at v_1. Consider rings R_m, $m = 0, 1, \ldots$ of width $a\ell(e_1)$ around u_1. R_m contains all links $e_k = (u_k, v_k) \in \mathcal{T}$ for which $ma\ell(e_1) \leq d(u_1, u_k) <$

$(m+1)a\ell(e_1)$ (cf. Figure 2). We know that any given link $e_k = (u_k, v_k) \in \mathcal{T}$, does not interfere with link $e_1 = (u_1, v_1)$. Therefore $\forall e_k \in \mathcal{T}, e_k \neq e$, we have $d(u_1, u_k) > a \max\{\ell(e_1), \ell(e_k)\}$. Therefore the first ring R_0 will not contain any other links from set $\mathcal{T}$, except for link e_1. The area of the ring R_m can be calculated as,

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

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

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

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

$$
\begin{aligned}
\mathcal{I}_m(v_1) &\leq 48m2^\alpha \frac{2J(u_1)}{(am\ell(e_1))^\alpha} \\
&= 2^\alpha \frac{96J(u_1)}{a^\alpha m^{\alpha-1}\ell(e_1)^\alpha}.
\end{aligned}
$$

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

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

Therefore the SINR at receiver v_1 is

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

where the second inequality above follows from the condition that $a^\alpha \geq 2^\alpha \frac{96\beta(\epsilon+1)}{\epsilon(\alpha-2)}$. The last expression above is at least β if $J(u_1) \geq (1+\epsilon)\beta N\ell(e_1)^\alpha$, which is ensured in the path selection and power control stage. Therefore, the schedule $\mathcal{S}$ produced by procedure Schedule is valid. $\square$

Length of the schedule

We shall now prove that the length of the schedule obtained by procedure Schedule is a polylog-factor away from that of the optimal schedule length. We first obtain an upper bound on the number of colors used by the greedy coloring scheme (steps 15 through 18), then by applying the Chernoff bounds, we show that the probability of congestion on any link being high is very low. Therefore the upper bound obtained on the length of the schedule holds with high probability.

LEMMA 3. $|\mathcal{S}'| \leq C + D$.

Lemma 3 is straightforward; the proofs of Lemma 4 and Lemma 5 are provided in the appendix.

LEMMA 4. *The length of the partial schedule $\mathcal{S}_t$ produced by procedure* Schedule *in step 9 is* $O(\max_e |C(e) \cap E_t| \log \Delta \log \Gamma)$.

LEMMA 5. *For each $t = 1, \ldots, |\mathcal{S}'|$,*

$$Pr[\max_e \mathcal{C}_t(e) \geq 4 \log \max\{n, C + D\}] \leq \frac{1}{n(C + D)}$$

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

LEMMA 6. $|\mathcal{S}|$ *is* $O((C + D) \log n \log \Delta \log \Gamma)$*, with probability at least $1 - \frac{1}{n}$.*

PROOF. Applying the union bound along with Lemma 5 and Lemma 3 we get

$$Pr[\exists\, t : \max_e \mathcal{C}_t(e) \geq 4 \log \max\{n, C + D\}] \leq \frac{C + D}{n(C + D)},$$

i.e., at most $1/n$. Thus, $|\mathcal{S}_t| \leq 4 \log \max\{n, C + D\} \log \Delta \log \Gamma$ for all t, with probability at least $1 - \frac{1}{n}$, which implies that the length of $\mathcal{S}$ is at most $O((C+D) \log n \log \Delta \log \Gamma)$ with this probability. □

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

Length of Optimal Schedule

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

LEMMA 7. *Consider any schedule $\mathcal{S}_{OPT}(p_{min}, (1 - \epsilon)p_{max})$ of optimal length for the given problem, that uses power levels from the range $[p_{min}, (1 - \epsilon)p_{max}]$. Then, $|\mathcal{S}_{OPT}(p_{min}, (1 - \epsilon)p_{max})| = \Omega((C_{OPT} + D_{OPT})/ \log \Gamma)$, where $C_{OPT} + D_{OPT}$ denotes the smallest congestion plus dilation that is possible if all paths have links $e = (u, v)$ such that $p_{max}(1 - \epsilon) \geq \beta N d(u, v)^\alpha$ for some constant ϵ.*

PROOF. Consider an optimal set of paths $\mathcal{P}_{OPT}$ that has congestion and dilation equal to C_{OPT} and D_{OPT}, respectively. Let $G_i = \{e_k = (u_k, v_k) \in E : 2^i \leq J(u_k) < 2^{i+1}\}$, for $i = 1, \ldots, \log \Gamma$. We now fix any $e = (u, v)$ and any time step t. Let $A_t = \{e_j = (u_j, v_j) : j = 1, \ldots, r\}$ be the set of links in $C(e)$ that are scheduled in $\mathcal{S}_{OPT}(p_{min}, (1 - \epsilon)p_{max})$ at time t. Further let $H_{t,i} = \{e_k \in A_t \cap G_i\}$. We first argue that the number of links that can be simultaneously scheduled from the set $H_{t,i}$ for any edge e, at any time t and any $i \in \{1, \ldots, \log \Gamma\}$ is $O(1)$. Let the links in set $H_{t,i}$ be arranged in the non-decreasing order of their lengths. Let there be s links in set $H_{t,i}$. Therefore $d(u_1, v_1) \leq d(u_2, v_2) \leq \ldots \leq d(u_s, v_s)$. Since all these links are used simultaneously, the SINR at each node v_j should exceed β. We shall compute the SINR at receiver v_s. Consider a link $e_j = (u_j, v_j) \in H_{t,i}$ (refer to figure 3). Since link $e_j, e_s \in C(e)$, according to the definition of $C(e)$, we know that $d(u, u_j) \leq ad(u_j, v_j)$ and $d(u, u_s) \leq ad(u_s, v_s)$. Further since $e_j = (u_j, v_j), e_s = (u_s, v_s) \in G_i$, we have $\frac{J(u_s)}{2} \leq J(u_j) \leq 2J(u_s)$.

It can be seen that

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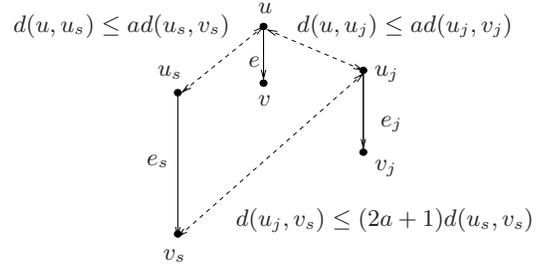&\leq (2a + 1)d(u_s, v_s).
\end{aligned}
$$

Figure 3: For a given link $\mathbf{e} = (\mathbf{u}, \mathbf{v})$ and set of other links $\mathbf{H_{t,i} = \{e_s, e_j\} \in C(e) \cap G_i}$ scheduled by $\mathcal{S}_{\mathbf{OPT}}(\mathbf{p_{min}}, (\mathbf{1 - \epsilon})\mathbf{p_{max}})$ at time $\mathbf{t}$, $\mathbf{d(u_j, v_s) \leq (2a + 1)d(u_s, v_s)}$

The interference experienced by v_s due to all such e_j is

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

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

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

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

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

5. ANALYSIS OF THE ALGORITHM

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

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

PROOF. From Lemma 6, we know that the maximum length of the schedule produced is $O((C + D) \log n \log \Delta \log \Gamma)$, from Theorem 1 we have $C + D = O((C_{OPT} + D_{OPT}) \log n)$ and from Lemma 7 we have $|\mathcal{S}_{OPT}(p_{min}, (1 - \epsilon)p_{max})| = \Omega((C_{OPT} + D_{OPT})/\log \Gamma)$. Therefore by putting everything together we have end-to-end latency of at most $O(\log^2 n \log \Delta \log^2 \Gamma \mathcal{S}_{OPT}(p_{min}, (1-\epsilon)p_{max}))$. □

The bounds achieved for algorithm MinDelay are worst-case approximation bounds. The algorithm complexity depends on (i) number of nodes (n) in the network and (ii) constants Δ, Γ. As mentioned earlier, δ denotes the maximum inter-point separation. and Γ denotes the ratio between maximum and minimum power assigned to nodes belonging to set V. It should be noted that the current implementation of algorithm MinDelay is best suited for moderately dense and closely confined ad hoc networks. For ad hoc networks that are widely spread and dense (containing large number of nodes), the inter-point separation between nodes could be high. Further since the power assigned is proportional to the edge lengths, the values of δ and Γ could be fairly high. The worst-case running time of our algorithm for such cases could increase. The performance of our algorithm for such cases would be worth exploring.
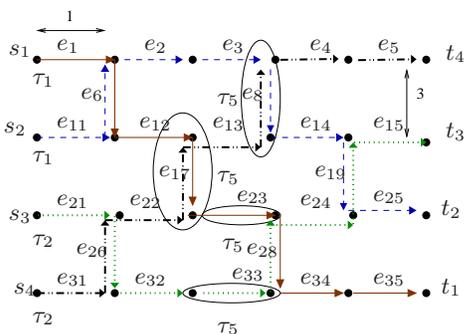
6. ILLUSTRATIVE EXAMPLE

Figure 4: Invalid Schedule: Due to random delays, sources s_1,s_2 start at time τ_1 and s_3, s_4 start at time τ_2. At time τ_5, links $e_8, e_{17}, e_{23}, e_{33}$ interfere with each other as $C(e_8) = e_{17}, C(e_{17}) = e_8, C(e_{23}) = e_{17}, C(e_{33}) = e_{17}$. This leads to an invalid schedule .

We briefly discuss an illustrative example showing how the algorithm works. Consider the network in Figure 4. Let the routes from s_i to t_i be as shown in the figure using corresponding dotted colored lines. For the sake of simplicity, let us assume that there is uniform power assignment and all nodes have been assigned power J. Let the random delays assignment be such that, source s_1 starts at time τ_1, s_2 at τ_1, s_3 at τ_2 and s_4 at τ_2. It can be seen from Figure 4 that packets on links $e_8, e_{17}, e_{23}, e_{33}$ are scheduled at the same time τ_5. Therefore $E_5 = \{e_8, e_{17}, e_{23}, e_{33}\}$. Since $\ell(e_8) = \ell(e_{17}) = 3$ and $\ell(e_{23}) = \ell(e_{33}) = 1$. According to Definition 1, if $a = 2$, we have $e_{17} \in C(e_8), e_8 \in C(e_{17}), e_{17} \in C(e_{23}), e_{17} \in C(e_{33})$. These links therefore interfere with each other and the schedule is indeed invalid. In this example, there are many such instances of invalid schedules. We therefore need to convert this invalid schedule into a valid schedule. According to step 7 of the procedure PathSelection, we now partition the links in set E_5 in two sets, such that $B_0 = \{e_{23}, e_{33}\}, B_1 = \{e_{17}, e_8\}$. Since we have assumed uniform power assignment, we can ignore the partitioning in step 13 of the procedure PathSelection. We now color the links in individual sets using the greedy coloring procedure described in steps 15 through 19. It can be verified that links e_{23}, e_{33} are scheduled at time τ' and links e_{17}, e_8 are scheduled at time $\tau' + 1, \tau' + 2$ respectively. The final schedule can be formed by pasting together these small sub-schedules.

7. COROLLARIES AND EXTENSIONS

We now sketch techniques for solving additional problems that are variations/extensions of our CLM problem.

End-to-end delay minimization with joint routing and scheduling for fixed power levels: This is a variation of the CLM problem wherein the power levels are fixed. Algorithm MinDelay could be used to solve this problem, with a few modifications. The constraints (5) which ensure that power assigned is valid, need to be removed from the LP formulation in procedure PathSelection. Procedure PowerControl is not required.

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

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

8. CONCLUSION AND FUTURE WORK

We described a general algorithmic technique leading to efficient polynomial-time centralized approximation algorithms for minimizing end-to-end latency by jointly considering routing, scheduling and power control layers. The results extend recent work of [14, 21, 22] by simultaneously considering multiple layers in the stack and more realistic models of radio interference. A number of questions remain open. First, our algorithms are centralized and thus cannot be used to design *distributed* cross-layer protocols for these problems. Second, we can further improve the algorithm to efficiently handle random packet arrivals. Third, two other performance-metrics of particular interest are rate throughput and fairness. It would be interesting to investigate if our approach can be adapted to obtain provable algorithms for optimizing these metrics as well.

9. REFERENCES

[1] P. Bjorklund, P. Varbrand, and D. Yuan, A Column Generation Method for Spatial TDMA Scheduling in Ad Hoc Networks, *Ad Hoc Networks*, vol. 2, Issue 4, pp. 405-418, 2004.

[2] R. Bhatia, and M. Kodialam, On Power Efficient Communication over Multi-hop Wireless Networks: Joint Routing, Scheduling and Power Control, *IEEE INFOCOM*, vol. 2, pp. 1457- 1466, March 2004.

[3] G. Brar, D. Blough, and P. Santi, Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks, *ACM MOBICOM*, pp. 2-13, 2006.

[4] L. Chen, S. Low, M. Chiang, and J. Doyle, Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks, *IEEE INFOCOM* , pp. 1-13, April 2006.

[5] L. Chen, S. Low, M. Chiang, and J. Doyle, Jointly Optimal Congestion Control, Routing, and Scheduling for Wireless Ad Hoc Networks, *IEEE INFOCOM* April 2006.

[6] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures, To appear *Proceedings of IEEE,* vol. 95, Issue 1, pp. 255-312, January 2007.

[7] H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. *Annals of Mathematical Statistics*, vol. 23, Number 4, pp. 493–509, 1952.

[8] R. L. Cruz, and A. V. Santhanam, Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks, *IEEE INFOCOM*, pp. 702-711, March 2003.

[9] T. ElBatt, and A. Ephremides, Joint Scheduling and Power Control for Wireless Ad-hoc Networks, *IEEE INFOCOM*, vol. 2, pp. 976-985, June 2002.

[10] P. Gupta, and P. Kumar, The Capacity of Wireless Networks, *IEEE Transactions on Information Theory*, vol. 46, Issue 2, pp. 388-404, March 2000.

[11] D. Knuth, The Art of Computer Programming, *Vol. 3 - Sorting and Searching*, Addison-Wesley (1973).

[12] M. Kodialam, and T. Nandagopal, Characterizing Achievable Rates in Multi-hop Wireless Networks: The Joint Routing and Scheduling problem, *ACM MOBICOM* , pp. 42 - 54, September 2003.

[13] G. Kulkarni, V. Raghunathan, and M Srivastava, Joint End-to-End Scheduling, Power Control and Rate Control in Multi-hop Wireless Networks, *IEEE GLOBECOM* , pp. 3357-3362, December 2004.

[14] V.S. Anil Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan, End-to-end Packet-Scheduling in Wireless Ad-Hoc Networks, *SODA*, pp. 1021-1030, 2004.

[15] F. Kuhn, T. Moscibroda, and R. Wattenhofer, Unit Disk Graph Approximation, *ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC),* October 2004.

[16] T. Leighton, B. Maggs, and S.Rao, Packet Routing and Job Shop Scheduling in O(Congestion+Dilation) steps, *Combinatorica*, vol. 14, Issue 2, pp. 167-180, 1994.

[17] X. Lin, and N. Shroff, Joint Rate Control and Scheduling in Multihop Wireless Networks, *43rd IEEE Conference on Decision and Control, Paradise Island,* vol. 2, pp. 1484- 1489, December 2004.

[18] X. Lin, and N. Shroff, The Impact of Imperfect Scheduling on Cross-layer Rate Control in Multihop Wireless Networks, In *IEEE INFOCOM*, vol. 3, pp. 1804- 1814, March 2005.

[19] X. Lin, N. Shroff, and R. Srikant, A Tutorial on Cross-Layer Optimization in Wireless Networks, *IEEE Journal on Selected Areas in Communications on Non-Linear Optimization of Communication Systems*, vol. 24, Issue 8, pp. 1452- 1463, June 2006.

[20] T. Moscibroda, R. Wattenhofer and Y. Weber, Protocol Design Beyond Graph-Based Models, *5th Workshop on Hot Topics in Networks (HotNets)*, November 2006.

[21] T. Moscibroda, and R. Wattenhofer, The Complexity of Connectivity in Wireless Networks, *IEEE INFOCOM* , pp. 1-13, April 2006.

[22] T. Moscibroda, R. Wattenhofer, and A. Zollinger, Topology Control Meets SINR: The Scheduling Complexity of Arbitrary Topologies, *ACM MOBIHOC* , pp. 310-321, 2006.

[23] M. Neely, Optimal Energy and Delay Tradeoffs for Multi-User Wireless Downlinks, *IEEE INFOCOM*, pp. 1-13, April 2006.

[24] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs, *Journal of Computer and System Sciences*, vol. 37, Issue 2, pp. 130-143, October 1988.

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

[26] S. Ramanathan, A Unified Framework and Algorithm for (T/F/C) DMA Channel Assignment in Wireless Networks, *Proc. IEEE INFOCOM* , pp. 900-907, April 1997.

[27] S. Ramanathan, and E. Lloyd, Scheduling Algorithms for Multi-Hop Radio Networks, *IEEE/ACM Transactions on Networking*, vol. 1, pp. 166-172, 1993.

[28] A. Srinivasan, and C. Teo, A Constant Factor Approximation Algorithm for Packet Routing and Balancing Local vs. Global criteria, *SIAM Journal on Computing*, vol. 30, pp. 2051-2068, 2001.

[29] V. Vazirani, Approximation Algorithms, Springer-Verlag, Berlin, 2001.

[30] L. Xiao, M. Johansson, and S. Boyd. Simultaneous Routing and Resource Allocation via Dual Decomposition. *IEEE Transactions on Communications*, vol. 52, Issue 7, pp. 1136-1144, July 2004.

10. APPENDIX

Proof for Lemma 4: Procedure Schedule schedules the edges in E_t by putting together schedules $\mathcal{S}_{t,i}$, $i = \log \Delta, \ldots, 0$,which in turn is formed by putting together schedules $\mathcal{S}_{t,i,j}$, $j = \log \Gamma, \ldots, 0$ which forms a schedule of the edges in $E_t \cap B_i \cap W_i^j$. We will argue below that for each $j = 0, \ldots, \log \Gamma$, the edges in $E_t \cap B_i \cap W_i^j$ can be scheduled in time $O(\max_e |C(e) \cap E_t \cap B_i|)$; the Lemma then follows. Consider the coloring process in steps 15 through 19 of the algorithm. Let $E_t \cap B_i \cap W_i^j = \{e_1, \ldots, e_s\}$, with $\ell(e_1) \geq \ell(e_2) \geq \ldots \geq \ell(e_s)$. The algorithm chooses a color for each e_k in this order - the color chosen for e_k is the smallest numbered color that is not used by $C(e_k) \cap \{e_1, \ldots, e_{k-1}\}$. Therefore, the number of colors used is at most $\max_e |C(e) \cap E_t \cap B_i| + 1$. Since there are $\log \Gamma$ such sets and $\log \Delta$ sets for each of the $\log \Gamma$ sets, the lemma follows.

Proof for Lemma 5: Fix any time t. For each edge $e \in E$, we define $Q(e) = \{q_i : e \in \mathcal{P}_i\}$ to be the set of packets passing through $C(e)$ in the schedule $\mathcal{S}'$. Consider any edge e. Let Y_i be an indicator variable, that is 1 if packet $q_i \in Q(e)$ crosses some $e' \in C(e)$ at time t in $\mathcal{S}'$. Then, $\sum_{q_i \in Q(e)} Y_i = |C(e) \cap E_t|$. Let e be the lth edge in path P_i of packet q_i. Then, $Y_i = 1$ if and only if $X_i = t - l$. Therefore, $Pr[Y_i = 1] \leq \frac{1}{C}$, and $E[\mathcal{C}_t(e)] = E[\sum_{q_i \in Q(e)} Y_i] = \sum_{q_i \in Q(e)} E[Y_i] \leq 1$ Since the random delays are chosen independently, the variables Y_i are independent Bernoulli trials and by the Chernoff bound, we have $Pr[\mathcal{C}_t(e) \geq \omega.E[\mathcal{C}_t(e)]] \leq 2^{-\omega}, \forall \omega \geq 6$. Therefore,
$$Pr[\max_e \mathcal{C}_t(e) \geq \omega] \leq Pr[\bigvee_{e \in E} \mathcal{C}_t(e) \geq \omega]$$
$$\leq \sum_{e \in E} Pr[\mathcal{C}_t(e) \geq \omega]$$
$$\leq \frac{|E|}{2^\omega} \leq \frac{n^2}{2^\omega}.$$
Therefore, $Pr[\max_e \mathcal{C}_t(e) \geq 4 \log \max\{n, C + D\}] \leq \frac{1}{n(C+D)}$ for $\omega = 4 \log \max\{n, C + D\}$.