Approximation Algorithms for Throughput Maximization in Wireless Networks with Delay Constraints

Guanhong Pei, *Student Member, IEEE,* Srinivasan Parthasarathy, *Member, IEEE,*
Aravind Srinivasan, *Fellow, IEEE,* and Anil Kumar S. Vullikanti, *Member, IEEE*

***Abstract*—We study the problem of throughput maximization in multi-hop wireless networks with end-to-end delay constraints for each session. This problem has received much attention starting with the work of Grossglauser and Tse (2002), and it has been shown that there is a significant trade-off between the end-to-end delays and the total achievable rate. We develop algorithms to compute such trade-offs with provable performance guarantees for arbitrary instances, with general interference models. Given a target delay-bound $\Delta(c)$ for each session c, our algorithm gives a stable flow vector with a total throughput within a factor of $O\big(\frac{\log \Delta_m}{\log\log \Delta_m}\big)$ of the maximum, so that the per-session (end-to-end) delay is $O\big((\frac{\log \Delta_m}{\log\log \Delta_m}\Delta(c))^2\big)$, where $\Delta_m = \max_c\{\Delta(c)\}$; note that these bounds depend only on the delays, and not on the network size, and this is the first such result, to our knowledge.**

***Index Terms*—Approximation Algorithms, Cross-layer Design, End-to-end Delay, Optimization, Throughput.**

I. Introduction

The end-to-end delay is an important issue in many multi-hop wireless network applications, such as video streaming [27], and there is a trade-off between the total achievable throughput and the delays; an important open question in this area has been to decide if it is possible to achieve delays proportional to the number of hops for each session, without much loss in throughput or throughput region [22]. Here we study the problem of computing explicit throughput-delay trade-offs in arbitrary networks. Given a multi-hop wireless network represented by a graph $G = (\mathcal{V}, \mathcal{L})$ and a set of sessions with a target delay $\Delta(c)$ for each session c, the goal of the *Delay-constrained Throughput Maximization* (DCTM) problem is to find a stable rate vector $\lambda()$ that (approximately) maximizes the total achievable rate $\sum_c \lambda(c)$, while ensuring that the average per-packet delay for session c is at most $\Delta(c)$ as possible. This problem is NP-hard, even without considering any delay guarantees [43], [4]; however, with delay constraints, this problem becomes hard to solve even approximately, as we establish later. Therefore, we study "bi-criteria" approximation algorithms which approximately maximize the total throughput, while allowing the delay constraints to be relaxed by a certain factor; our focus is on designing algorithms with provable approximation guarantees.

While there has been much work on delay-throughput trade-offs, especially for random networks or restricted 1-hop sessions, the best results for average end-to-end delay bounds known so far considering scheduling with fixed routes and traffic rates are by [25], [22], [24], [28]. Kar et al. [25] study both max-weight and randomized independent set scheduling policies, and show that the average packet delay is bounded by the chromatic number $C(\mathcal{N})$ of the interference graph; $C(\mathcal{N})$ represents the minimum number of independent sets into which the link conflict graph of a network $\mathcal{N}$ can be partitioned. Jagabathula and Shah [22] design a scheduling scheme that ensures per-session end-to-end delays of $O\left(\#hops\right)$ with the total throughput within a constant factor of the optimum; however, this result is restricted to primary interference, whereas for general graph-based interference, the delay bound becomes $O\left(\#hops \cdot D^2\right)$, where D denotes the maximum degree in the conflict graph (which could be high). Let n and m denote the number of nodes and edges (links) respectively, in the given network. Jayachandran and Andrews [24] design a scheduling scheme that ensures per-session end-to-end delay of $O\left(\#hops \cdot n\right)$. Le et al. [28] prove that max-weight scheduling has a network-average delay bound of $O\left(\#hops_{max} \cdot \theta_{max} \cdot n \cdot m\right)$, where θ_{max} is the maximum number of flows going through any link in the network. The bounds in prior results more or less depend on the network size, and obtaining per-flow delay bounds independent of the network size had been an important open problem. In this paper, we develop an algorithm for DCTM under the unit disk graph model based on our scheduling scheme that improves on the average delay bounds under the general graph-based interference model.

Our main contributions are as follows.

1. *Approximation hardness of* DCTM. We show lower bounds on the computational complexity of the DCTM prob-

The work of Guanhong Pei and Anil Kumar S. Vullikanti has been partially supported by NSF Nets Grant CNS-0626964, NSF PetaApps Grant OCI-0904844, DTRA R&D Grant HDTRA1-0901-0017, DTRA CNIMS Grant HDTRA1-07-C-0113, NSF NETS CNS-0831633, DOE DE-SC0003957, and NSF Netse CNS-1011769. A. Srinivasan's work has been supported in part by NSF ITR Award CNS-0426683, NSF Award CNS-0626636, and NSF Award CNS 1010789. A preliminary version of this paper appears in the Proceedings of IEEE INFOCOM, Shanghai, China, April 2011.

Guanhong Pei was with the Department of Electrical and Computer Engineering, and Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061 USA, during the time of this work (e-mail: guanhong@vt.edu).

Srinivasan Parthasarathy is with IBM T.J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: spartha@us.ibm.com).

Aravind Srinivasan is with the Department of Computer Science, and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 USA (e-mail: srin@cs.umd.edu).

Anil Kumar S. Vullikanti is with the Department of Computer Science, and Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: akumar@vbi.vt.edu).

lem. When the wireless network is modeled as a unit-disk graph, we show that there is a constant K such that it is NP-hard to approximate the DCTM problem within a factor of K; for arbitrary graphs DCTM is hard to approximate within a factor of $O(n^{1-\varepsilon})$ for any constant $\varepsilon \in (0, 1)$, while satisfying all delay constraints. In light of these hardness results, we need to relax the delay constraints: we study bi-criteria algorithms that simultaneously approximately maximize throughput and guarantee the delay performance.

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

Our result involves two basic steps: (i) Under the general interference model, when routes and mean traffic rates are fixed, we develop a sequence of *queueing system reductions* and *isolation* technique (detailed in Section V) to derive the end-to-end per-flow delay upper-bounds for our random-access scheduling scheme, such that the upper-bounds depend only on a flow's path length, while achieving a throughput region of $\frac{1}{eI_{max}}$. In contrast, previous results (*e.g.*, [25], [22], [24], [28]) depend on the network size or interference degrees. This has motivated the design and techniques in our rounding-based algorithm for traffic control and routing. (ii) We use a rounding-based algorithm for a linear-programming (LP) relaxation of the problem to construct a flow vector that uses "short" paths (or has "low" costs), to send "high" flow on each selected path. Our algorithm is based on a novel application of the Lovász Local Lemma [41], combined with filtering and path refining steps in order to reduce certain kinds of dependencies. Our specific rounding scheme is crucial in ensuring that the factor of loss in throughput is only $O\left(\frac{\log\Delta_m}{\log\log\Delta_m}\right)$; in contrast, a straight-forward and direct application of randomized rounding [31] can only lead to an $O(\log n)$ factor. Path-constrained flows have been studied in a wired setting, *e.g.*, [5], [19]; interference constraints and the fact that we need both short paths and large flow values makes our problem different.

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

The main focus of our paper is theoretical. The initial steps of computing the rate vector are centralized (though the scheduling is distributed), and our results are on static networks; nevertheless, our techniques give an efficient method to provide provable conservative delay-throughput trade-offs in any specific network, which can be useful in choosing suitable rate vectors: *e.g.*, as in the video streaming application of [27]. We also present in Appendix B one application of our framework and simulation in networks with adaptive channels (*e.g.*, in cognitive networks). We quantify the impact of adaptive channel switching on the throughput capacity and end-to-end delay.

Organization. We discuss related work in Section II and the network model and relevant definitions in Section III. We prove the approximation hardness of DCTM in Section IV. The delay bounds for a given rate vector $\lambda()$ are derived in Section V (Theorem 1), and in Section VI we describe our algorithm for computing a good rate vector; Theorem 2 shows the combined throughput and delay guarantees. We discuss extensions to asynchronous random-access schemes in Section VII, and to multi-channel networks in Appendix B. Finally, we describe our experimental results in Section VIII and conclude the paper in Section IX.

II. Related Work

Characterizing the rate region has been broadly studied in all kinds of wired and wireless networks. Here we only focus on the ones addressing throughput-delay issues.

Wired networks: A large class of papers, *e.g.*, [34], [13], [3], [29], [30], provide analytical guarantees on end-to-end delays and network utilization achievable through specific scheduling protocols in multi-hop *wired* networks. However, none of these explicitly deal with the problem of routing to simultaneously guarantee network utilization and end-to-end delays.

Random wireless networks: Precise trade-offs between the network capacity and end-to-end delay as well as other parameters such as fairness, or number of radio channels (in a multi-radio multi-channel network) have been well studied for wireless networks under the assumption that the physical node locations follow *uniform spatial distributions*. Building on [18], [15], El Gamal et al. [12] show the relationship between average delay and (per-node) capacity. This problem has been extended in various directions, *e.g.*, [33], [12], [42], [38], [39]. However, in general, the techniques employed analyzing random wireless networks do not help shed light on the delay-throughput trade-offs in an arbitrary wireless network (with non-random topology).

Arbitrary wireless networks: The design and analysis of wireless protocols for arbitrary networks (with non-random topologies) from the perspective of guaranteeing network utilization and end-to-end delays is relatively less-understood; some of the recent papers on this topic include [25], [22], [24], [28] discussed in Section I. Besides, there are also important delay results for *single-hop* traffic [23], [32], [26], [17]. Jaramillo, Srikant, Ying [23] propose a scheduling algo-

TABLE I: Comparison of relevant scheduling and delay bound results for arbitrary wireless networks (for given traffic and flow routes with general interference). Notation used: n: #nodes; m: #links; I_{max}: max interference degree; θ_{max}: max congestion (#flows through a link); $C(\mathcal{N})$: chromatic number of link interference graph; $C^{\mathcal{M}}(\mathcal{N})$: chromatic number of the interference graph based on flow-links [25].

Type	In Paper	Delay Bound	Scheduling Scheme	Efficiency Ratio	Type of Delay Bound
End-to-end	[22]	$O\left(\#hops \cdot D^2\right)$	preemptive LIFO & stable marriage	$\Omega\left(1/D^2\right)$	per-session, upper
	[24]	$O\left(\#hops \cdot n\right)$	coordinated EDF or max-weight	$\Theta(1)$	per-session, upper
	[28]	$O\left(\#hops_{max} \cdot \theta_{max} \cdot n \cdot m\right)$	max-weight	$\Theta(1)$	network-average, upper
	[25]	$O\left((\#hops_{max})^2 \cdot C^{\mathcal{M}}(\mathcal{N})\right)$	max-weight indep. set	$\Theta(1)$	network-average, upper
	Ours	$O\left((\#hops)^2\right)$	random access	$\Omega\left(1/I_{max}\right)$	per-session, upper
	[17]	non-analytical	max-weight & variants	$\Theta(1)$	network-average, lower
Single-hop	[32]	$O(1)$	maximal scheduling	$\Omega\left(1/I_{max}\right)$	network-average, upper
	[26]	$O\left(C(\mathcal{N})\right)$	max-weight or randomized indep. set	$\Theta(1)$	network-average, upper

rithm to satisfy long-term QoS requirements under heterogeneous delay constraints of a specific class of periodic traffic. Neely [32] shows that general maximal matching policies achieve $O(1)$ network-average delay for given traffic. Kar, Luo and Sarkar [26] show that the maximum expected delay depends linearly on the chromatic number of the interference graph. Gupta and Shroff [17] give an algorithm for computing lower bounds for average delay under max-weight scheduling. We compare our scheduling and delay results with recent relevant results in Table I. We note that, our *unique* DCTM framework maximizes throughput with low per-session delay guarantees that only depend on target delays.

Based on random-access scheduling policies, there has been a lot of research studying end-to-end delay. However, most of them use approximate models with tools from statistical mechanics, such as diffusion-approximation model [7], totally asymmetric simple exclusion process [40], and approximate queueing model [45], [37]; it is not clear to what extent these simplified *approximate* statistical queueing models can be applied for the multi-hop dynamics in general networks with general arrival processes. We provide analytical bounds without simplifying either queueing or traffic model.

The throughput and delays depend on constraints from multiple layers, e.g., routing and scheduling. Recently Huang et al. [21] propose a cross-layer framework that incorporates congestion control and scheduling constraints to deal with delays. Their sophisticated algorithm is well designed to achieve order-optimal delays with a throughput region comparable to that of maximal scheduling. While their framework is similar to ours, their work does not consider routing, and does not handle explicitly given per-session bounds.

In summary, the main aspects which distinguish our work from the prior results include: (1) We provide a theoretical analysis and competitive performance guarantees of the end-to-end delay and the achievable throughput, in arbitrary network with general traffic; (2) our work can also serve a way of guiding practical network design and management by finding rate vectors, choosing paths, scheduling to achieve specific delay-throughput guarantees.

III. Network Model and Scheduling

A. Network Model

The wireless network is modeled as a directed graph $G = (\mathcal{V}, \mathcal{L})$. A link $(u, v) \in \mathcal{L}$ denotes that u can transmit to v directly. We use a general graph-based interference model as in [32], [16]. Interference is defined by means of *interference sets*: each link l has its own interference set $\mathcal{I}(l) \triangleq \{l' : l' \text{ interferes with } l\}$. We define *maximum interference degree*, denoted by I_{max}, as the maximum number of links that can transmit simultaneously within any $\mathcal{I}(l)$; in most interference models, I_{max} remains a constant. We consider *bidirectional* interference, which accounts for ACK packets that may be crucial in a distributed setting with random-access. Our derivation of the delay bounds applies to general interference models. For results in Section VI, we assume a unit-disk model [36], in which each node u has a fixed transmission range (assumed to be 1, w.l.o.g.), and $(u, v) \in \mathcal{L}$ if and only if $d(u, v) \le 1$; two links interfere when one end of a link is within h hops from one end of the other, where h is a constant integer.

TABLE II: Summary of notation used in the paper.

G	network graph	$\mathcal{I}(l)$	interference set of l
$\mathcal{V}$	set of nodes	$\mathcal{L}_{in}(v)$	incoming links of v
$\mathcal{L}$	set of links	$\mathcal{L}_{out}(v)$	outgoing links of v
n	#nodes	$\mathcal{C}$	set of connections
μ	service rate	Λ^{OPT}	capacity region
λ	mean arrival rate	A	exogenous arrival
$s(c)$	source node of c	$t(c)$	destination node of c
Q	queue or backlog	p_l	channel access prob. of l
a	arrival rate	d	actual # departure pkts
Λ	throughput region	$\mathcal{F}$	set of flows
$\Delta(c)$	target delay of c	OPT	max total throughput
$\chi(G)$	chromatic number	I_{max}	max interference degree

Time is divided into uniform and contiguous *slots* of length 1. Define $\mu_l(t) \in \{0, 1\}$ as the service rate for link l at time slot t; $\mu_l(t)$ is determined by the specific scheduling protocol used. For simplicity, we use time and link service models similar to [32], [26], [17]; our results can be extended to cases where link capacity is more than 1.

Let $\mathcal{C}$ denote the set of connections or sessions. Let $s(c)$ and $t(c)$ denote the source and destination, respectively, for session $c \in \mathcal{C}$. Each session c might use multiple paths (also referred to as flows, in this paper) for communication; let $\mathcal{F}(c)$ denote the set of paths/flows that can be used by session c. Let $\mathcal{L}(f)$ denote the set of links on flow f. Let $A_f(t)$ denote the exogenous arrival process for flow f. We use i_f, where $i > 0$ is an integer, to denote the ith link on f (*e.g.,* 1_f is the 1st link in $\mathcal{L}(f)$). We assume the exogenous arrival process of each flow to be i.i.d over time and independent of each other;

the first moment $\mathbb{E}\{A_f(t)\} = \lambda(f)$ and the second moment $\mathbb{E}\{A_f^2(t)\} \leq A^{(2)}$, where $A^{(2)}$ is a constant.

In our queueing model, each link l is associated with a queue Q_l, which holds packets waiting for transmission. Q_l is divided into logical FIFO sub-queues for each flow through l. Let $Q_{l,f}$ denote the logical FIFO sub-queue for flow f on link l. We also use the same notation to denote the backlogs: For each time slot t, let $Q_l(t) \triangleq \sum_{f\in\mathcal{F}(l)} Q_{l,f}(t)$ denote the queue size, where $Q_{l,f}(t)$ denotes the number of packets waiting for transmission on link l for flow f, and where $\mathcal{F}(l)$ is the set of flows that include l in their paths[1]. Each time l is activated to transmit, only one logical queue on l gets serviced. Each packet of flow f traverses from $Q_{1_f,f}$ through $Q_{|\mathcal{L}(f)|_f,f}$. For a slot t, let $a_{i_f,f}(t)$ be the number of arrival packets for f at $Q_{i_f,f}$, $d_{i_f,f}(t)$ the actual number of f's departure packets from $Q_{i_f,f}$, and $\mu_{i_f,f}(t)$ the service rate offered to $Q_{i_f,f}$. We assume the transmission takes place during the entire slot, and at the end of the current slot, the arrival to a queue is counted in the backlog. The queue-evolution mechanism can be expressed as

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

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

B. Throughput Region, Delay and Problem Definition

A schedule $\mathcal{S}$ in our model describes the times at which data is moved over the links of the network. A scheduling scheme is said to be *stable* if the average delay is bounded and consequently, all backlogs have bounded sizes. Formally,

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

The *throughput region* Λ^S of a scheduling scheme S is the closure of the set of all exogenous arrival rate vectors that can be stably supported under S. The *network capacity region* Λ^{OPT} is the closure of the set of all rate vectors that can be stably supported by any feasible scheduling scheme; it is known that max-weight scheduling can stably schedule any traffic vector interior to Λ^{OPT} (but without any delay constraints) [14].

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

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

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

C. Random-access Scheduling

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

$$p_l(t) = 1 - \exp\left(e \sum_{f:\, l\in\mathcal{L}(f)} \lambda(f)/(1-\epsilon)\right), \tag{2}$$

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

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

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

$$\sum_{l'\in\mathcal{I}(l)\cup\{l\}} \sum_{f:\, l'\in\mathcal{L}(f)} \lambda(f) \leq \frac{1-\epsilon}{e},\ \forall l \in \mathcal{L}. \tag{3}$$

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

It follows from [9] that the above constraints in Inequality (3) with RHS scaled up by eI_{max} are necessary conditions for *any* stable scheduling scheme; note that I_{max} is usually a constant in most interference models. Inequality (3) defines a throughput region within a factor of $1/(eI_{max})$ of Λ^{OPT}. In other words, this random-access gives an $O(I_{max})$ approximation of the capacity region comparable to that of maximal scheduling, while maintaining a low-complexity distributed manner of operation.

IV. Approximation Hardness of DCTM

DCTM is NP-hard: this follows from the fact that even without any delay constraints, the throughput maximization problem is NP-hard [43]. We extend this to show that even simple cases of the DTCM problem are hard even to approximate, if the delay bounds are required to be satisfied; this motivates the need for bi-criteria approximations.

Lemma 1: There is a constant $K' > 0$ such that the DCTM problem cannot be approximated within a factor of K' if the interference graph G is a unit-disk graph, unless $P = NP$.

Proof: Our proof is based on the result of Clark et al. [10], which shows that given a unit-disk graph $G = (\mathcal{V}, \mathcal{L})$ (where $\mathcal{V}$ is the nodes set and $\mathcal{L}$ is the link set), it is NP-complete to distinguish between the case $\chi(G) = 3$ and $\chi(G) = 4$, where $\chi(G)$ is the chromatic number of G. We reduce this to an instance of DCTM in an interference model based on distance-2 independence model (*i.e.*, two transmissions are simultaneously possible only if the senders are at least distance-2 apart): let $\mathcal{V}'$ be a duplicate node set of $\mathcal{V}$, such that each $v \in \mathcal{V}$ and its counterpart v' in $\mathcal{V}'$ are located very close; let $G'' = (\mathcal{V}'', \mathcal{L}'')$ be a graph where $\mathcal{V}'' = \mathcal{V} \cup \mathcal{V}'$, and $\mathcal{L}'' = \mathcal{L} \cup \{(v, v') : v \in \mathcal{V}\}$. For each node $v \in \mathcal{V}$, we construct n sessions originating at v and ending at each node v'. For each session c, let the target delay $\Delta(c) \triangleq 3$. We assume that the exogenous traffic is at a constant bit-rate for each session c. If $\chi(G) = 3$, observe that a throughput rate of $1/3$ for each connection is possible — within each window of three time steps, all the connections can be scheduled in this interference model, making the total throughput of $n/3$ feasible. On the other hand, if $\chi(G) = 4$, at most $K'n$ of the sessions can be colored using 3 colors (this follows from a simple analysis of the reduction of [10]). Since $\Delta(c) = 3$ for each session, at most $K'n$ sessions can be scheduled within a window of size 3, implying a total throughput of at most $K'n/3$. ■

For arbitrary interference graphs, it can be shown by a reduction from graph coloring that DCTM cannot be approximated within a factor of $O(n^{1-\varepsilon})$, for any $\varepsilon \in (0, 1)$; we omit the proof because of space limitations.

V. Delay Upper-bounds

We now derive end-to-end delay bounds for flows with a given feasible average rate vector $\lambda()$ that satisfies the constraints in Inequality (3). The main idea is to progressively "isolate" each flow as a tandem system for which we can easily find a bound on the per-flow delay.

Theorem 1: For a rate vector $\lambda()$ that satisfies Inequality (3), the random-access scheduling protocol described in Section III-C ensures that (i) the system is stable, implying a throughput region of $\frac{\Lambda^{OPT}}{eI_{max}}$; (ii) the average delay for each flow f is $O\left(|\mathcal{L}(f)|^2/(\lambda(f))^2\right)$; and (iii) the average network delay is $O\left(\sum_{f\in\mathcal{F}} |\mathcal{L}(f)|^2 \Big/ \left(\sum_{f\in\mathcal{F}} \lambda(f) \min_{f\in\mathcal{F}}\{\lambda(f)\}\right)\right)$.

We start the proof with the following lower bound on the expected service rate $\mu_{l,f}(t)$ for any flow f and link l; this will be used in all our analysis in the rest of this section. For notational simplicity, we use $x(l, f) = \lambda(f)/(1-\epsilon)$, and $x(l) = \sum_{f\in\mathcal{F}(l)} x(l, f)$. Rewriting Inequality (3), we have $\sum_{l'\in\mathcal{I}(l)\cup\{l\}} x(l') \le 1/e$. Then, Equation (2) can be rewritten as $p_l(t) = 1 - e^{-ex(l)}$, and that gives us:

$$\begin{aligned}
\mathbb{E}\{\mu_{l,f}(t)\} &\ge p(l,f)p_l(t) \prod_{l'\in\mathcal{I}(l)} (1 - p_{l'}(t)) \\
&\ge p(l,f)p_l(t) \sum_{l'\in\mathcal{I}(l)} e^{-ex(l')} \\
&\ge p(l,f)\left(1 - e^{-ex(l)}\right) e^{ex(l)-1} \\
&= \frac{\left(e^{ex(l)} - 1\right) x(l,f)}{ex(l)} \ge x(l,f).
\end{aligned} \tag{4}$$

The idea for the proof of the above theorem is that due to the properties of random-access scheduling, each flow can be viewed in "isolation" as a tandem system, with lower bounds on the expected service rate of $\mu(l, f)$ that only depend on $x(l, f)$ for each logical queue $Q_{l,f}$, as shown in Equation (4). Let the triplet $(Q(), a(), \mu())$ denote a queueing system. From now on till the end of this section, we use R to denote the basic queueing system under the basic scheduling scheme specified in Section III-C, with the queueing model and the exogenous arrival processes described in Section III-A. We put R at superscript to denote the quantities of the corresponding system.

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

Lemma 2: Let R' and R'' be two identical queueing systems (with the same initial states and the same set of general i.i.d. arrival processes) but only differ in the service rates: for R', the service rate at time t for the ith link of flow f is $\mu^{R'}_{i_f,f}(t) \in \{0, 1\}$, and for R'' it is $\mu^{R''}_{i_f,f}(t) \in \{0, 1\}$; $\mathbb{E}\left\{\mu^{R'}_{i_f,f}(t)\right\} \ge \mathbb{E}\left\{\mu^{R''}_{i_f,f}(t)\right\}$, for each flow f and link i_f, at any time t. Then, for each flow f, the average total queue size in R' is no greater than that in R'', *i.e.*,

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

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

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

where $\varepsilon(f) = \epsilon\lambda(f)/(1-\epsilon)$; the exogenous arrival rates remain the same. Then, Lemma 2 implies that $\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R_1}_{i_f,f}(\tau)\right\} \geq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R}_{i_f,f}(\tau)\right\}$. Note that whether the reduced system is using wireless medium no longer matters.

(2) **Reduction 2**. Note that the exogenous arrival at the source link of the tandem system $(Q^{R_1}(), a^{R_1}(), \mu^{R_1}())$ is a rather general arrival process, making it nontrivial to use earlier methods directly, *e.g.*, [6], [11], [20], in bounding the end-to-end delays. We seek to reduce the system R_1 to another queueing system R_2 so that the arrival process for each non-source queue is also Bernoulli. The queueing system R_2 is defined in the following manner: at each time slot t the service rate for each link i_f in $\mathcal{L}_f$ on flow f is the same as that for R_1, except that this link tries to access the medium even if the queue $Q^{R_2}_{i_f,f}$ is empty or does not have enough packets to fill in the capacity. In case (i_f, f) gets serviced and $Q^{R_2}_{i_f,f}$ has a backlog smaller than the channel capacity, dummy packets are injected to make full use of the channel capacity during the time slot. These packets will be labeled as packets for flow f. Now that we have unit capacities, each time (i_f, f) accesses the medium, it transmits one packet to the next queue in line. Therefore, the service process of $Q^{R_2}_{i_f,f}$, and the arrival process at the subsequent queue $Q^{R_2}_{(i+1)_f,f}$ coincide. Note that if the number of retransmission is upper-bounded, the arrival at non-source queues will be smaller. In the system R_1, for any i, $\mu^{R_1}_{i_f,f}$ follows a Bernoulli distribution, which implies that the subsequent arrival is also a Bernoulli process with $a^{R_2}_{(i+1)_f,f}(t) = \mu^{R_2}_{i_f,f}(t) = \mu^{R_1}_{i_f,f}(t)$. This leads to Lemma 3, which directly follows from Lemma 2:

Lemma 3: For each flow f,

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

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

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

Next, we perform Lyapunov drift analysis to derive an upper-bound on the queue size of each $Q^{R_2}_{i_f,f}$. Refer to [14], [32] for the details of this approach. We define the Lyapunov function as $\mathbb{L}\left(Q^{R_2}_{i_f,f}(t)\right) \triangleq \left(Q^{R_2}_{i_f,f}(t)\right)^2$.

The 1-step Lyapunov drift is then defined as:

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

By referring to Equation 1 and Lemma 4.3 of [14], we obtain

$$\begin{aligned}
&\triangle^{(1)}_{Q}\left(Q^{R_2}_{i_f,f}(t)\right) \\
\leq&\mathbb{E}\left\{\left(\mu^{R_2}_{i_f,f}(t)\right)^2 + \left(a^{R_2}_{i_f,f}(t)\right)^2 \middle| Q^{R_2}_{i_f,f}(t)\right\} - \\
&\mathbb{E}\left\{2Q^{R_2}_{i_f,f}(t)\left(\mu^{R_2}_{i_f,f}(t) - a^{R_2}_{i_f,f}(t)\right) \middle| Q^{R_2}_{i_f,f}(t)\right\}. \qquad (5)
\end{aligned}$$

Further, $\mathbb{E}\left\{\left(\mu^{R_2}_{i_f,f}(t)\right)^2\right\} = \mathbb{E}\left\{\mu^{R_2}_{i_f,f}(t)\right\} \leq 1/e$, and $\mathbb{E}\left\{\mu^{R_2}_{i_f,f}(t) - a^{R_2}_{i_f,f}(t)\right\} = \varepsilon(f)/|\mathcal{L}(f)|$. Additionally, for $i = 1$, we have $\mathbb{E}\left\{\left(a^{R_2}_{1_f,f}(t)\right)^2\right\} \leq A^{(2)}$; when $i > 1$, $\mathbb{E}\left\{\left(a^{R_2}_{i_f,f}(t)\right)^2\right\} = \mathbb{E}\left\{a^{R_2}_{i_f,f}(t)\right\} \leq 1/e$. Inequality (5) can be then rewritten as

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

We use $\overline{Q}_{i_f,f}$ to denote $\limsup\limits_{t\to\infty}\dfrac{1}{t}\sum\limits_{\tau=0}^{t-1}\mathbb{E}\left\{Q_{i_f,f}(\tau)\right\}$ for simplicity. From Theorem 1 in [32] which deduces an inequality from the Lyapunov drift, $\overline{Q}^{R_2}_{i_f,f} \leq \frac{1+\max\left\{1, eA^{(2)}\right\}}{2e\varepsilon(f)}|\mathcal{L}(f)|$. Therefore, the sum of mean backlogs for flow f is

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

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

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

The average network delay is

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

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

VI. Multi-commodity Flows with Delay Guarantees

From Theorem 1, it follows that the average end-to-end delay bound for flow f is proportional to $|\mathcal{L}(f)|^2$, and inversely proportional to $\lambda^2(f)$; the average network end-to-end delay bound is in proportion to $\sum_{f\in\mathcal{F}}|\mathcal{L}(f)|^2$, and inversely in proportion to $\sum_{f\in\mathcal{F}}\lambda(f)$ and $\min_{f\in\mathcal{F}}\lambda(f)$. Therefore, in order to find a feasible rate vector $\lambda()$ that minimizes the delay guarantees, we need to construct flows with "high" rate (*i.e.*, to keep $\lambda(f)$ high) and "short" paths (*i.e.*, to make $|\mathcal{L}(f)|$ low). In this section, we use terms "path" and "flow" interchangeably.

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

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

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

Corollary 1: Using the rate vector $\lambda()$ along with the random-access scheduling scheme described in Section V, we ensure that, for each session c and each flow $f\in\mathcal{F}''(c)$, the average delay is $O\left((\frac{\log\Delta_m}{\log\log\Delta_m}\Delta(c))^2\right)$.

Our algorithm involves construction of multi-commodity flows with constraints on the paths used; broadly, these constraints bound the sum of the "costs" of the links on the paths, which will be explained later with the LP formulation. Our approach employs an approximation algorithm, which selectively drops some cost-unfavorable sessions and maximizes the rates of the rest of the sessions. Path constrained flows have been studied in a wired setting, *e.g.*, [5], [19], but the interference constraints, and the fact that we need both short paths and large flow values make our problem different and difficult. Our algorithm is comprised of the following 3 steps: (1) linear programming formulation, (2) path filtering, and (3) randomized rounding.

(Step 1) **Linear programming formulation**. We use the LP formulation as a basis to compute an upper-bound of $OPT(\Delta())$, and to develop our bi-criteria approximation algorithm. For session c, recall that $\mathcal{F}(c)$ denotes the set of possible paths from $s(c)$ to $t(c)$. We assume a cost function defined on the links; let $cost(l)$ denote the cost of link l. For path f, we define $cost(f)=\sum_{l\in\mathcal{L}(f)}cost(l)$ as the cost of path f. The costs can be defined in a fairly general manner. In most of this paper, the cost of a link l will denote the time (in number of slots) needed for a packet transmission (ignoring queuing and interference delays); therefore, the cost of a path will be proportional to its length. Let $\mathcal{F}(c,L)$ denote the set of paths of cost at most L from $s(c)$ to $t(c)$. As mentioned in Section III, we will assume that all link capacities are 1. We start with the following LP formulation (LP) to find a flow vector $y()$ that maximizes $\sum_c y(c)$ subject to Constraints LP-(6a) to LP-(6e). Here, $y(c)$ denotes the total rate for connection c, and $y(f)$ the rate along path $f\in\mathcal{F}(c)$; $y(l,c)=\sum_{f\in\mathcal{F}(c):l\in\mathcal{L}(f)}y(f)$ is the total flow for c along l.

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

$$\text{s.t.}\quad \forall c,\ y(c)=\sum_{f\in\mathcal{F}(c)}y(f) \tag{6a}$$

$$\forall c,\ \sum_{f\in\mathcal{F}(c)}y(f)cost(f)\le\Delta(c)y(c) \tag{6b}$$

$$\forall l,c,\ y(l,c)=\sum_{f\in\mathcal{F}(c):\ l\in\mathcal{L}(f)}y(f) \tag{6c}$$

$$\forall l,\ \sum_{l'\in\mathcal{I}(l)\cup\{l\}}\sum_c y(l',c)\le\frac{1-\epsilon}{e} \tag{6d}$$

$$\forall f,\ y(f)\ge 0 \tag{6e}$$

In the above formulation: (1) Constraints LP-(6a) and LP-(6c) represent path-based flow-conservation constraints. (2) LP-(6b) constrains the total path cost, which we use as a lower-bound on the average delay along a flow-path; in our case the cost function is chosen to be path length since end-to-end delay is lower bounded by the number of hops. (3) Congestion constraints in LP-(6d) ensure the stability under a random-access scheme; Note that under the h-hop unit disk graph model, since the interference degree $I_{max}=O(h)=O(1)$, $\sum_{l'\in\mathcal{I}\cup\{l\}}\sum_c y(l',c)\le I_{max}=O(1)$ for any link l.

We write the optimal objective value of (LP) as $OPT^{LP}(\Delta())$. Since LP-(6b) serves as relaxed delay constraints and LP-(6d) may potentially compromise the optimal value of total flow by at most a constant factor, $OPT^{LP}(\Delta())=\Omega\big(OPT(\Delta())\big)$.

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

(1) replacing Constraints LP-(6a) and LP-(6c) with for all c, $\sum_{l\in\mathcal{L}_{out}(s(c))}y(l,c)=y(c)$ and $\sum_{l\in\mathcal{L}_{in}(t(c))}y(l,c)=y(c)$, and flow-conservation constraints at all other nodes;
(2) replacing LP-(6b) with for all c, $\sum_{l\in\mathcal{L}}y(l,c)cost(l)\le\Delta(c)y(c)$; and
(3) replacing LP-(6e) with for all l,c, $y(l,c)\ge 0$.

Let $y^*()$ denote the optimum fractional solution to the above LP; $OPT^{LP}(\Delta())=\sum_c y^*(c)$. Following standard techniques, *e.g.*, [2], this flow can be decomposed into path flows $y^*(f)$ in polynomial time, with a polynomial number of paths that have positive flow. Let $\mathcal{F}^*=\{f: y^*(f)>0\}$ be the set of flows with positive flow.

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

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

It follows by a simple averaging argument, that

$$\forall c,\ y'(c) = \sum_{f \in \mathcal{F}^*(c, 2\Delta(c))} y'(f) \geq y^*(c)/2. \tag{7}$$

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

(Step 3) **Randomized rounding**. In this step, we round the filtered solution to an integral solution to obtain Lemma 4.

Lemma 4: After the randomized rounding step, we obtain a set $\mathcal{F}'' \subseteq \mathcal{F}^*$ of paths with positive rates, and a rate vector $\lambda()$, such that (1) for each $f \in \mathcal{F}''$, $\lambda(f) = \Omega\left(\log\log\Delta_m/\log\Delta_m\right)$; (2) $\sum_c \lambda(c)$ is $\Omega\left(\log\log\Delta_m/\log\Delta_m\right) OPT^{LP}(\Delta())$; and (3) the chosen paths incur "low" congestion; more precisely, for each link l, we have $\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_c \lambda(l', c) \leq \frac{1-\epsilon}{e}$, where $\lambda(l', c)$ is the rate on all paths $f \in \mathcal{F}''(c)$ such that $l' \in \mathcal{L}(f)$.

We first describe the sub-steps of the rounding stage and then discuss the proof of Lemma 4.

(1) **Pre-processing:** we partition paths into groups, formulate a minimax integer program (MIP) that minimizes maximum congestion and that chooses one path in each group with "large" flow rate, and formulate a relaxation of the minimax integer program with refined paths.
(2) **Randomized rounding:** we employ the techniques based on [41] to derive an approximate solution to the relaxed minimax integer program.
(3) **Post-processing:** we scale down the flow rates by a "reasonably small" factor, such that the congestion constraints LP-(6d) can be satisfied.

The details are provided below.

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

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

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

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

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

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

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

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

$$\begin{aligned} \textbf{MIP:} \quad & \min \quad w \\ & \text{s.t.} \quad \sum_{f \in \mathcal{F}'_i} z(f) = 1,\ \forall i = 1, \ldots, k && \text{(8a)} \\ & \sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_{f \in \mathcal{F}' : l' \in \mathcal{L}(f)} z(f) \leq w,\ \forall l && \text{(8b)} \\ & z(f) \in \{0, 1\},\ \forall f \in \mathcal{F}' && \text{(8c)} \end{aligned}$$

(MIP) above formulates a minimax integer program that minimizes maximum congestion among all the interference sets. Constraints MIP-(8a) and MIP-(8c) let us choose one flow path from each set $\mathcal{F}'_i$, and assign flow rate of 1 to the paths chosen. Since $\sum_{f \in \mathcal{F}'_{max}} y'(f) = \Theta(1)$, the vector y', after suitable scaling is a feasible solution for the linear relaxation of (MIP); further, it will turn out that the objective value of an optimum fractional solution to (MIP) is $O(1)$. MIP-(8b) can be rewritten in an aggregate manner as $\mathbf{B}\vec{z} \leq \vec{w}$, where $\mathbf{B}$ is a $|\mathcal{L}| \times |\mathcal{F}'|$ matrix. Note that $|\mathcal{F}'|$ is polynomial in $|\mathcal{L}|$. Intuitively, this integer program is hard to solve exactly, because matrix $\mathbf{B}$ can be dense and irregular, as a result of the facts that an interference set can be as large as $\mathcal{L}$ and that both of the set of flow paths in an optimal solution and the graph topology are non-controllable. In light of this, we perform the following path refinement (Step 3.3), constraint relaxation (Step 3.4) and MIP reformulation (Step 3.5) to approximately solve (MIP). It is in these steps that we require an h-hop unit-disk graph (UDG) model as described in Section III.

(Step 3.3) **Pre-processing: Path refinement**. For each link $l = (u, v)$, if there is a path $f \in \mathcal{F}'$ that uses more than a constant number, K_0, of links in $\mathcal{I}(l)$, we "short-cut" f into f' that uses at most K_0 such links, and does not violate any of the constraints of (LP). This is illustrated in Figure 1.

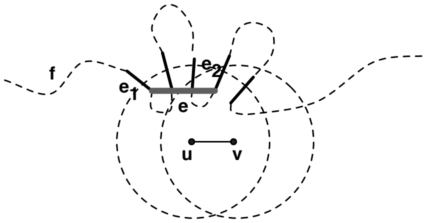


Fig. 1: Path refinement operation: Consider path f (shown by dashed curved line) and link $l = (u, v)$, in a UDG model. Path f revisits $\mathcal{I}(l)$ multiple times, and the segment of f from between the endpoints of edges e_1 and e_2 can be replaced by edge e (shown in light gray and bold) to get a path f' which is shorter; sending the same flow on f' (instead of f) is still feasible. We then continue to use $\mathcal{F}'$ to denote the refined set of paths without ambiguity.

(Step 3.4) **Pre-processing: Relaxation of congestion constraints**. Recall that we have rewritten MIP-(8b) into $\mathbf{B}\vec{z} \leq \vec{w}$. Let $\eta(\mathbf{B})$ denote the maximum number of congestion constraints (in MIP-(8b)) in which a path in $\mathcal{F}'$ is simultaneously involved; the definition is based on the fact that a path f appears in every congestion constraint corresponding to the links on f and those interfering with links on f. It is crucial that $\eta(\mathbf{B})$ be "small", so that the rounding scheme in later steps produces a "good" approximate ratio. However, $\eta(\mathbf{B})$ may become $\Omega\left(\Delta_m \max_l |\mathcal{I}(l)|\right)$ if we use the original set of congestion constraints in LP-(6d), which is the case for general interference model. To control this, in a UDG model, we now coarsen the formulation in the following manner:

(1) We partition the plane into $\frac{1}{8} \times \frac{1}{8}$ grid cells. A cell is said to be *non-empty* if there is at least one node in it. Let $\mathcal{B}$ denote the set of non-empty cells and let b denote a cell in $\mathcal{B}$. We say a link $l \in b$ if and only if l is within a constant distance of any point in cell b. Since the path refinement do not increase path lengths, all the path lengths are bounded by $2\Delta_m$. The number of cells that a path $f \in \mathcal{F}'$ goes through is hence $O\left(\Delta_m\right)$.

(2) MIP-(8b) implies the following:

$$\forall b \in \mathcal{B}, \sum_{l \in b} \sum_{f \in \mathcal{F}': l \in \mathcal{L}(f)} z(f) \leq w. \tag{9}$$

We scale the coefficients of any rate variable $z(f)$ in Inequality (9) down by a constant factor of K_0 (which upper-bounds the number of links of a path that lie in the same interference set (as a result of Step 3.3)), such that the coefficients in the inequality above falls in $[0,1]$. That renders a relaxed set of congestion constraints as

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

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

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

$$\begin{aligned} \textbf{MIP-1:} \quad & \min \quad w \\ & \text{s.t.} \quad \sum_{f \in \mathcal{F}'_i} x(f) = 1, \ \forall i = 1, \ldots, k && \text{(10a)} \\ & \qquad \mathbf{B}'\vec{x} \leq \vec{w} && \text{(10b)} \\ & \qquad x(f) \in \{0,1\}, \ \forall f \in \mathcal{F}' && \text{(10c)} \end{aligned}$$

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

(Step 3.6) **Rounding process**. Applying the rounding algorithm of [41] to solving (MIP-1) yields a rate vector $x'()$, such that a path f_i with rate $x'(f_i) = 1$ is chosen from $\mathcal{F}'$ for each group $\mathcal{F}'_i$ (except group $\mathcal{F}'_k$, in case $\sum_{f \in \mathcal{F}'_k} y'(f) < \sum_{f \in \mathcal{F}'_1} y'(f)$); and for the rest of the paths not chosen, zero flows are assigned.

Lemma 5: Let $\mathcal{F}'' = \{f : x'(f) > 0\}$ denote the set of selected paths with positive rates; $\mathcal{F}'' \subseteq \mathcal{F}'$. The rounding process ensures that (1) for each $f \in \mathcal{F}''$, $x'(f) = 1$; (2) for each connection c, $x'(c) = \sum_{f \in \mathcal{F}''(c)} x'(f)$. (3) $|\mathcal{F}''| = \sum_c x'(c) = \Theta(\sum_c y^*(c)) = \Theta\big(OPT^{LP}(\Delta())\big)$; and (4) for each link l, we have $\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_c x'(l', c) \leq K_1 \log \Delta_m / \log\log \Delta_m$, where $x'(l', c)$ is the flow rate on path $f \in \mathcal{F}''(c)$ such that $l' \in \mathcal{L}(f)$, and K_1 is a constant.

In [41], randomized rounding techniques are proposed for solving a minimax integer program (MIP-2) as shown below.

$$\begin{aligned} \textbf{MIP-2:} \quad & \min \quad w \\ & \text{s.t.} \quad \sum_{j \in \mathcal{X}_i} x_{i,j} = 1, \ \forall i \in \mathcal{X} && \text{(11a)} \\ & \qquad \mathbf{A}\vec{x} \leq \vec{w} && \text{(11b)} \\ & \qquad x_{i,j} \in \{0,1\}, \ \forall i \in \mathcal{X}, \ \forall j \in \mathcal{X}_i && \text{(11c)} \end{aligned}$$

(MIP-2) seeks to minimize maximum components in $\mathbf{A}\vec{x}$ while satisfying the equality constraints MIP-2-(11a) and the integrity constraints MIP-2-(11c). $\mathcal{X}$ is a set of distinct integers, and $\mathcal{X}_i$ is a set of distinct integers associated with integer $i \in \mathcal{X}$. The total number of $x_{i,j}$ variables is denoted by $N = \sum_i |\mathcal{X}_i|$. $\vec{x}$ denotes the N-dimensional vector of variables $x_{i,j}$. $\mathbf{A} \in [0,1]^{M \times N}$ is an $M \times N$ matrix, and $\vec{w}$ is an M-dimensional vector with variable w in each component.

Lemma 6 (Lemma 2.4(a) in [41]): Given independent r.v.s $X_1, \ldots, X_n \in [0,1]$, let $X = \sum_{i=1} X_i$ and $\mu = \mathbb{E}\{X\}$. Then, $\forall \mu > 0$, $\forall p \in (0,1)$, $\exists \delta = H(\mu, p) > 0$, such that $\lceil \mu\delta \rceil (e^{\delta}/(1+\delta)^{1+\delta})^{\mu} \leq p$, and such that

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

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

Theorem 3 (Theorem 2.5 in [41]): Let w^*_{LP} denote the optimum value of the LP relaxation of (MIP-2). Then there exists an integral solution of value at most $w^*_{LP} + O(1) + O(\min\{w^*_{LP}, M\} H(\min\{w^*_{LP}, M\}, 1/\gamma(\mathbf{A})))$ for (MIP-2).

proof of Lemma 5: We observe that (MIP-1) is an instance of (MIP-2). Recall that $\gamma(\mathbf{A})$ denotes the maximum number of non-zero entries in any column of $\mathbf{A}$; note that $\gamma(\mathbf{B}') = \eta(\mathbf{B}') = O\left(\Delta_m\right)$, and we will refer to this later. Since w^*_{LP} is the optimum value of the LP relaxation of (MIP-2), w^*_{LP} is a lowerbound on the optimum value of (MIP-2). According to Lemma 6 and Theorem 3, if (MIP-2) satisfies the three conditions —

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

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

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

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

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

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

(2) W.l.o.g., we assume at least one connection requires more than one hop. We construct a feasible solution x'_{LP} for the LP relaxation of (MIP-1) and show that the corresponding objective value w'_{LP} is at most 1. For each i and each $f \in \mathcal{F}'_i$, we assign $x'_{LP}(f) = y'(f)/\sum_{f' \in \mathcal{F}'_i} y'(f')$, such that constraints MIP-1-(10a) are satisfied. On the one hand, any path of two or more hops touches at least 6 cells; that implies $\gamma(\mathbf{B}') \geq 6$. On the other hand, because a path has at most K_0 links in any interference set, $\sum_{l \in b} \sum_{f \in \mathcal{F}': l \in \mathcal{L}(f)} y'(f) \leq K_0 \sum_{f \in \mathcal{F}'_1} y'(f)$; and since $\forall f \in \mathcal{F}', x'_{LP}(f) \leq y'(f)/\sum_{f' \in \mathcal{F}'_1} y'(f')$, the congestion incurred by $x'_{LP}()$ in each cell b is

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

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

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

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

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

That can be translated to

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

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

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

which implies that the congestion at each interference set is $O(h^2 \log \Delta_m/\log\log \Delta_m) = O(\log \Delta_m/\log\log \Delta_m)$. Hence, Lemma 5 holds. ■

(Step 3.7) **Post-processing: Scaling and choosing flow vector**. We choose a rate vector $\lambda()$ as $\lambda(f) = K_2 \frac{\log\log \Delta_m}{\log \Delta_m} x'(f)$, $\forall f \in \mathcal{F}''$, where K_2 is a constant, such that the congestion constraints in LP-(6d) are satisfied. Note that for some connections, multiple flows might be chosen (whereas for some connections, none would be chosen); for each ("original") connection c, define $\lambda(c) = \sum_{f \in \mathcal{F}''(c)} \lambda(f)$ as the total flow of c. $\sum_c \lambda(c) = \frac{\log\log \Delta_m}{\log \Delta_m} \Omega\big(OPT^{LP}(\Delta())\big)$. As discussed before, $OPT^{LP}(\Delta()) = \Omega\big(OPT(\Delta())\big)$. Now Lemma 4 follows, and our algorithm ends.

Combining the delay analysis in Section V, gives us Theorem 2 and Corollary 1 — the bi-criteria approximation — for DCTM problem.

VII. Asynchronous Random-access Scheduling

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

The work of [8] gives the following results. Let γ denote the maximum ratio between $T_{tx}(l)$ of any link l and $T_{tx}(l')$ of any hidden interfering link l' of l, i.e., $\gamma = \max_l \max_{l' \in \mathcal{I}_{hid}(l)} \frac{T_{tx}(l)}{T_{tx}(l')}$. For a system under asynchronous random-access scheduling, we set the channel access probability of each link l:

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

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

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

The expected service rate for queue $Q_{l,f}$ on l at any time t is lower-bounded as $\mathbb{E}\{\mu_{l,f}(t)\} \geq \lambda(f)/(1-\epsilon)$. Details can be found in [8]. By using the queueing reduction technique in Section V, we obtain a similar delay bounds as in Theorem 1. Plugging in the stability constraints to (LP) in Section VI, we obtain the throughput-delay guarantees as in Theorem 4.

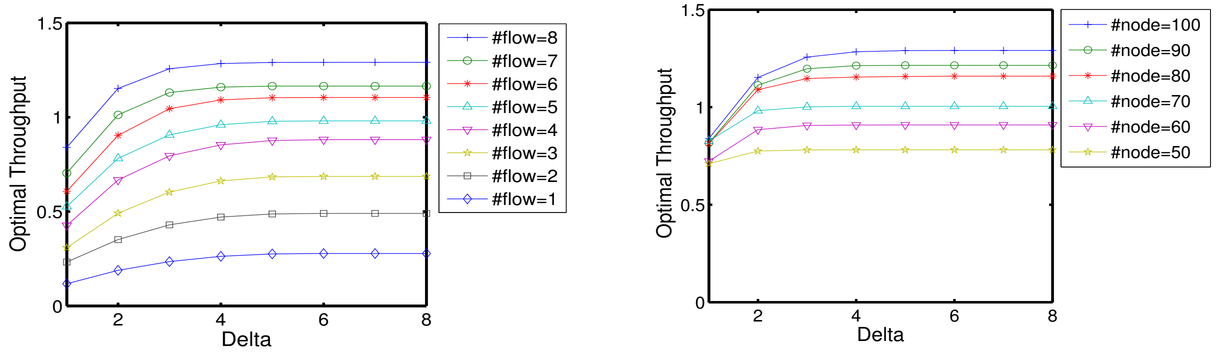
Theorem 4: (i) The rate vector $\lambda()$ resulted by our multi-commodity flow framework ensures that: $\sum_c \lambda(c) = \Omega\left(\frac{\log\log \Delta_m}{\gamma \log \Delta_m}\right) OPT\left(\Delta()\right)$. (ii) For each flow f in the set of flows $\mathcal{F}''(c)$ constructed for each session c if $\lambda(c) > 0$, the rate $\lambda(f) = \Omega\left(\frac{\log\log \Delta_m}{\gamma \log \Delta_m}\right)$, and the path has length at most $2\Delta(c)$. (iii) Using the random-access scheduling protocol, we ensure that, for each session c and each flow $f \in \mathcal{F}''(c)$, the average delay is $O\big((\frac{\gamma \log \Delta_m}{\log\log \Delta_m} \Delta(c))^2\big)$.

VIII. Simulation Results

We study the performance of our algorithms empirically with simulation. First, for single-channel models, we show how the optimal network throughput depends on varying uniform target delay, number of sessions and network size. Next, for a multi-channel network, we study the variation in the optimal throughput as the number of channels and Δ values vary. Experiments are carried out both on random unit-disk graph topologies and grid network topologies, with both primary interference and two-hop interference models. LP's are solved with SCIP [1] and SoPlex [44] bundle.

A. Single-channel Networks

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



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

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

Fig. 2: Trade-offs among OPT throughput, delay, number of flows, network size.

Figure 2 shows the throughput-delay trade-offs for different number of flows and different network size under the same interference model. Figure 2a features a fixed network size of 100, and Figure 2b features a fixed number of flows equal to 8. Intuitively, as Δ values increase, thereby loosening the delay constraint, the optimal throughput will rise; as the number of random connections goes up, throughput increases, since the optimization process gets more exploration space. The saturation of the curves happen where the interference plays a major role through the congestion constraints in the LP.

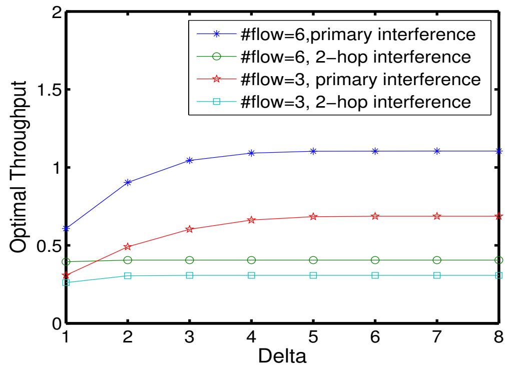


Fig. 3: Impact of interferences: primary interference and two-hop interference.

Figure 3 shows the impact of different levels of interference severance by using primary and 2-hop interference models, both on 100-node topologies. The increased interference severance (due to added flows and extended interference) causes the network throughput to reach a saturation point faster.

B. Distribution of Individual Flow Packet Delay

Figure 4 shows the cumulative distribution of individual flow packet delays. We conducted this set of experiments on 50-node network topologies with 5 flows that have i.i.d. arrivals and random paths, operating with a single channel. We run the simulation for 500 iterations. In Figure 4, we show the results for flows with path lengths (*i.e.*, number of hops) of 24, 16, 8, 7, 7 respectively. We notice that over 90% of the packets of each flow experience session delays linear in the number of hops. Further, the CDF curves are almost centro-symmetric within the 0.1% and 99.9% range, and the total span of the packet delays of each flow is small and is densely around the line in which over 90% of the packets of a flow attribute their delay distribution to. That means that with a high probability, all packet delays are subject to a reasonably small upper-bound. Moreover, the packet delay CDF for flows with the same path length tend to be the same, as can be seen from the two flows of lengths 7; we also have the same observations for flows of other lengths on other topologies (omitted here).

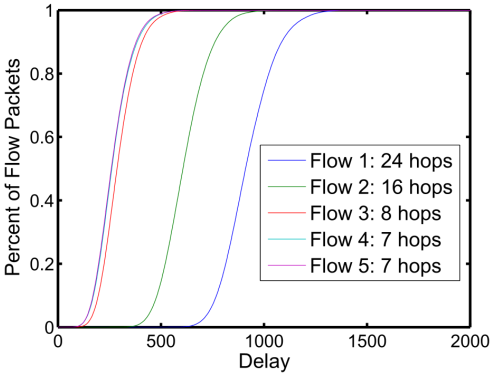


Fig. 4: CDF of individual flow packet delay.

In Figure 5, we show the average ratios of packet delay to path length for the flows with the same mean arrival rate. These CDF curves, which lie close to each other, share a small span of values and almost the same mean value, meaning that on a topology the flows that have the same arrival rate tend to share the same ratio of delay to path length under our random-access scheduling scheme. We also observe that the average individual packet delays tend to have a much lower growth rate than our analytical quadratic delay bound (which may be too conservative); that suggests that our random-access scheduling scheme may yield close-to-linear (or close-to-optimal) delay performance in practice.

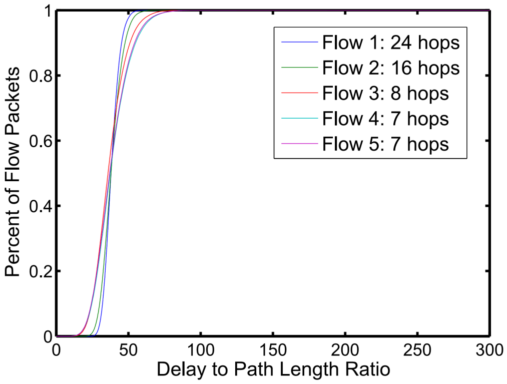


Fig. 5: CDF of individual packet delay to path length ratio.

C. Average Queue Size

With the same setting as that of the previous set of experiments in Section VIII-B, we observe the average backlogs for the following three representative types of flows: high-rate, medium-rate, and low-rate flows. The high-rate flow is close to saturating the stability condition in Inequality 3, while the medium-rate flow injects packets with half of the high rate, and the low-rate flow has an even smaller rate. Each of the flows has a path of at least 25 hops. Figure 6 shows the average queue sizes as a function of hop number. The average backlog of each flow stays much flatter than a quadratic curve in the number of hops; the higher the flow rate, the higher the backlog. We note that it is hard in general to prove a tight bound on average queue sizes and thereafter the average delays using Lyapunov drift techniques, which appears more efficient in the use of proving stability. For example, one can only get a delay bound exponential in path length by applying Lyapunov-drift-based analysis. Devising more efficient techniques stands as a major challenge to better understand scheduling in multi-hop wireless networks. It is necessary to develop problem-specific new analysis techniques towards this end, for example, in this paper, the queueing reduction technique to derive the quadratic delay bounds for the random-access scheduling scheme.

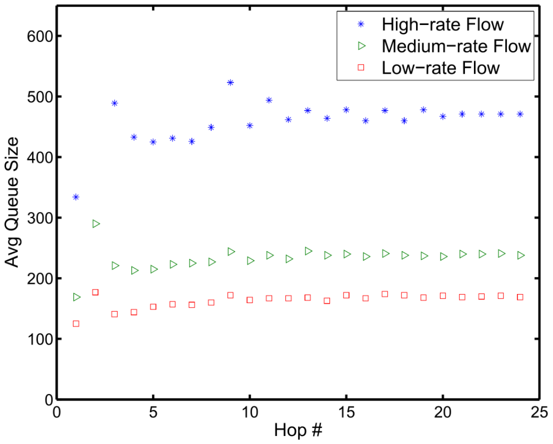


Fig. 6: Average queue size with no transmission threshold.

IX. Conclusion and Open Problems

Characterizing delay-throughput trade-offs and bounds is a fundamental problem in wireless networks, with numerous applications. In this paper, we develop a theoretical framework to rigorously bound this trade-off and provably approximate the maximum throughput with given per-session delay requirements. The performance guarantees we prove are worst-case bounds (in terms of problem instances), and are likely to be better for specific instances. Extending these techniques to bounding the average per-session delay with additional fairness constraints is a very challenging open problem. Maximal scheduling [32] and random-access maximal scheduling [16] also fit in our DCTM framework. However, tightly bounding per-session end-to-end delay in general is an open and difficult problem. New progress on this will likely enable our DCTM framework to work on a larger class of scheduling schemes. Besides, a potential application of our framework lies in the setting of physical interference based on SINR constraints. For example, it would be interesting to derive end-to-end delay bounds with an efficient scheduling scheme such as the random-access algorithm in [35].

Acknowledgements. We are grateful to the reviewers and for their valuable comments, which have helped improve the quality of this paper.

References

[1] T. Achterberg, “Scip: Solving constraint integer programs,” *Mathematical Programming Computation*, vol. 1, no. 1, pp. 1–41, 2009.

[2] R. Ahuja, R. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[3] M. Andrews and L. Zhang, “Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule,” in *IEEE Infocom*, 1999.

[4] ——, “Satisfying arbitrary delay requirements in multihop networks,” in *IEEE Infocom*, 2008.

[5] G. Baier, T. Erlebach, A. Hall, E. Köhler, H. Schilling, and M. Skutella, “Length-bounded cuts and flows,” in *ICALP (1)*, 2006, pp. 679–690.

[6] A. S. Bedekar and M. Azizoglu, “On the information-theoretic capacity of discrete-time queues,” *IEEE Trans. on Info. Theory*, vol. 44, pp. 446–461, 1998.

[7] N. Bisnik and A. Abouzeid, “Queuing network models for delay analysis of multihop wireless ad hoc networks,” *Ad Hoc Networks*, vol. 7, no. 1, 2009.

[8] D. Chafekar, D. Levin, V. S. A. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan, “Capacity of asynchronous random-access scheduling in wireless networks,” in *IEEE Infocom*, 2008.

[9] P. Chaporkar, K. Kar, and S. Sarkar, “Throughput guarantees through maximal scheduling in multi-hop wireless networks,” *IEEE Trans. on Info. Theory*, vol. 54, pp. 572–594, 2008.

[10] B. Clark, C. Colbourn, and D. Johnson, “Unit disk graphs,” *Discrete Mathematics*, vol. 86, pp. 165–177, 1990.

[11] M. Draief, J. Mairesse, and N. O’Connell, “Queues, stores, and tableaux,” *Journal of Applied Probability*, vol. 42, no. 4, pp. 1145–1167, 2005.

[12] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, “Throughput-delay trade-off in wireless networks,” in *IEEE Infocom*, 2004.

[13] L. Georgiadis, R. Guerin, V. Peris, , and K. Sivarajan, “Efficient network qos provisioning based on per node traffic shaping,” in *IEEE Infocom*, 1996.

[14] L. Georgiadis, M. J. Neely, and L. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” *Foundations and Trends in Netw.*, vol. 1, no. 1, pp. 1–149, 2006.

[15] M. Grossglauser and D. N. C. Tse, “Mobility increases the capacity of ad hoc wireless networks,” *IEEE/ACM Trans. on Netw.*, vol. 10, pp. 477–486, 2002.

[16] A. Gupta, X. Lin, and R. Srikant, “Low-complexity distributed scheduling algorithms for wireless networks,” *IEEE/ACM Trans. on Netw.*, vol. 17, no. 6, pp. 1846–1859, 2009.

[17] G. R. Gupta and N. B. Shroff, “Delay analysis for multi-hop wireless networks,” in *IEEE Infocom*, 2009.

[18] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[19] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis, “Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems,” in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, 1999, pp. 19–28.

[20] J. Hsu and P. Burke, “Behavior of tandem buffers with geometric input and markovian output,” *IEEE Trans. on Comm.*, vol. 24, no. 3, pp. 358–361, 1976.

[21] P.-K. Huang, X. Lin, and C.-C. Wang, “A low-complexity congestion control and scheduling algorithm for multihop wireless networks with order-optimal per-flow delay,” in *IEEE Infocom*, 2011.

[22] S. Jagabathula and D. Shah, “Optimal delay scheduling in networks with arbitrary constraints,” in *ACM SIGMETRICS*, 2008.

[23] J. J. Jaramillo, R. Srikant, and L. Ying, “Scheduling for optimal rate allocation in ad hoc networks with heterogeneous delay constraints,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, 2011.

[24] P. Jayachandran and M. Andrews, “Minimizing end-to-end delay in wireless networks using a coordinated edf schedule,” in *IEEE Infocom*, 2010.

[25] K. Kar, S. Sarkar, A. Ghavami, and X. Luo, “Delay guarantees for throughput-optimal wireless link scheduling,” *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2906–2911, 2012.

[26] K. Kar, X. Luo, and S. Sarkar, “Asymptotically tight delay guarantees through wireless link scheduling,” in *IEEE Infocom*, 2009.

[27] S. Kompella, S. Mao, Y. Hou, and H. Sherali, “On path selection and rate allocation for concurrent video sessions in ad hoc networks,” *IEEE/ACM Trans. on Netw.*, vol. 17, no. 1, pp. 212–224, 2009.

[28] L. B. Le, K. Jagannathan, and E. Modiano, “Delay analysis of maximum weight scheduling in wireless ad hoc networks,” in *Annual Conference on Information Sciences and Systems (CISS)*, 2009, pp. 389–394.

[29] C. Li and E. Knightly, “Coordinated network scheduling: A framework for end-to-end services,” in *IEEE ICNP*, 2000.

[30] ——, “Schedulability criterion and performance analysis of coordinated schedulers,” in *ITC*, 2001.

[31] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.

[32] M. J. Neely, “Delay analysis for maximal scheduling with flow control in wireless networks with bursty traffic,” *IEEE/ACM Trans. on Netw.*, vol. 17, no. 4, pp. 1146–1159, 2009.

[33] M. J. Neely and E. Modiano, “Capacity and delay tradeoffs for ad-hoc mobile networks,” *IEEE Trans. on Info. Theory*, vol. 51, pp. 1917–1937, 2005.

[34] A. K. Parekh and R. G. Gallager, “A generalized processor sharing approach to flow control in integrated services networks: The multiple node case.” *IEEE/ACM Trans. on Netw.*, vol. 2, no. 2, 1994.

[35] G. Pei and V. A. Kumar, “Low-complexity scheduling for wireless networks,” in *ACM Mobihoc*, 2012.

[36] S. Ramanathan and E. L. Lloyd, “Scheduling algorithms for multihop radio networks,” *IEEE/ACM Trans. on Netw.*, vol. 1, pp. 166–177, 1993.

[37] S. Ray, D. Starobinski, and J. Carruthers, “Performance of wireless networks with hidden nodes: A queueing-theoretic analysis,” *Journal of Computer Communications*, vol. 28, pp. 1179–1192, 2005.

[38] G. Sharma and R. Mazumdar, “Scaling laws for capacity and delay in wireless networks with random mobility,” in *IEEE ICC*, 2004.

[39] G. Sharma, R. Mazumdar, and N. B.Shroff, “Delay and capacity tradeoffs in mobile ad hoc networks: a global perspective,” *IEEE/ACM Trans. on Netw.*, vol. 15, no. 5, pp. 981–992, 2007.

[40] S. Srinivasa and M. Haenggi, “Taseps: A statistical mechanics tool to analyze the performance of wireless line networks,” submitted to IEEE/ACM Trans. on Netw., 2009.

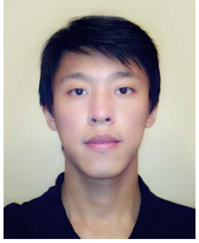
[41] A. Srinivasan, “An extension of the Lova´sz Local Lemma, and its applications to integer programming,” *SIAM J. Comput.*, vol. 36, no. 3, pp. 609–634, September 2006.

[42] S. Toumpis and A. Goldsmith, “Large wireless networks under fading, mobility, and delay constraints,” in *IEEE Infocom*, 2004, pp. 609–619.

[43] P.-J. Wan, “Multiflows in multihop wireless networks,” in *MobiHoc*, 2009, pp. 85–94.

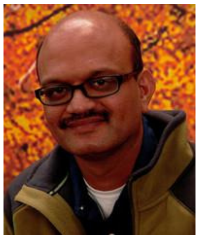
[44] R. Wunderling, “Paralleler und objektorientierter Simplex-Algorithmus,” Ph.D. dissertation, Technische Universität Berlin, 1996.

[45] M. Xie and M. Haenggi, “Towards an end-to-end delay analysis of wireless multihop networks,” *Ad Hoc Networks*, vol. 7, no. 5, pp. 849–861, 2009.



Guanhong Pei has a Ph.D. (2012) and an M.S. (2009) in Electrical and Computer Engineering (ECE) from Virginia Tech, Blacksburg, Virginia, USA. Before that, he received his B.S. in Electronics and Information Engineering and second-major bachelor’s degree in Business Administration (2006) from Shanghai Jiao Tong University (SJTU), Shanghai, China. He worked as research interns respectively at Alcatel-Lucent Bell Labs, New Jersey in 2010 and at DOCOMO USA Labs, California in 2008. Guanhong’s interest includes analysis and design of distributed algorithms, randomized algorithms, combinatorial optimization, and cross-layer system design.

Srinivasan Parthasarathy is a Research Staff Member at IBM T. J. Watson Research Center since 2006. Prior to joining IBM, he received his PhD in Computer Science from University of Maryland, College Park. His research interests are in the areas of algorithm design and optimization, networking, Big Data technologies, and cyber security. He has published in several top venues in Computer Science including JACM, IEEE/ACM ToN, IEEE FOCS, ACM-SIAM SODA, ACM Sigmetrics, ACM Sigmod, and IEEE Infocom.



Aravind Srinivasan (Fellow, IEEE) is a Professor (Dept. of Computer Science and Institute for Advanced Computer Studies) at the University of Maryland, College Park. He received his degrees from Cornell University (Ph.D.) and the Indian Institute of Technology, Madras, India (B.Tech.). His research interests are in algorithms, randomness, and networks, with applications in networking, social networks, health, and energy. He has published more than 90 papers in these areas, in journals including Nature, Journal of the ACM, IEEE/ACM Transactions on Networking, and SIAM Journal on Computing. He is an editor of three journals, and has served on the program committees of various conferences. He is a AAAS Fellow.

Anil Kumar S. Vullikanti is an Associate Professor (Dept. of Computer Science and the Virginia Bioinformatics Institute) at Virginia Tech. Prior to this, he was a technical staff member at the Los Alamos National Lab. He received B.Tech (1993) in Computer Science and Engineering from the Indian Institute of Technology, Kanpur, India, and the Ph.D. (1999) in Computer Science from the Indian Institute of Science, Bengaluru, India. His research interests include approximation algorithms, mobile computing, combinatorial optimization, and simulation of large socio-technical systems.