Approximation algorithms for throughput maximization in wireless networks with delay constraints

Guanhong Pei*, V. S. Anil Kumar†, Srinivasan Parthasarathy‡, and Aravind Srinivasan§
*Dept. of Electrical and Computer Engineering and Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061
†Dept. of Computer Science and Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, VA 24061
‡IBM T.J. Watson Research Center, Hawthorne, NY 10532
§Dept. of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742

***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 tradeoff between the end-to-end delays and the total achievable rate. We develop algorithms to compute such tradeoffs 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 $O\left(\frac{\log \Delta_m}{\log\log \Delta_m}\right)$ of the maximum, so that the per-session (end-to-end) delay is $O\left((\frac{\log \Delta_m}{\log\log \Delta_m}\Delta(c))^2\right)$, 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.**

I. Introduction

The end-to-end delay is an important issue in many multi-hop wireless network applications, such as video streaming [25], and there is a tradeoff between the total achievable throughput and the delays; an important open question in this area has been to decide if it is possible to achieve delays proportional to the number of hops for each session, without much loss in throughput or throughput region [21]. In this paper, we study the problem of computing explicit throughput-delay tradeoffs in specific given networks. Given a multi-hop wireless network represented by a graph $G = (\mathcal{V}, \mathcal{L})$, 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 maximizes the total achievable rate $\sum_c \lambda(c)$, while ensuring that the per-packet delay for session c is as close to $\Delta(c)$ as possible. This problem is $\mathcal{NP}$-Complete, even without considering any delay guarantees [43], [4]; however, with delay constraints, this problem becomes hard to solve even approximately, as we discuss later. Therefore, in this paper, we study "bi-criteria" approximation algorithms, which maximize the total throughput, while allowing the delay constraints to be violated by some factor; our focus is on designing algorithms with provable approximation guarantees.

While there has been a lot of work on delay-throughput tradeoffs, especially for random networks or restricted 1-hop sessions, the best bounds for end-to-end delays known so far are by [21], [22], [27]. Jagabathula and Shah [21] design a scheduling scheme that ensures per-session end-to-end delays of $O\left(\#hops\right)$ with the total throughput within a constant factor of the optimum; however, this result is restricted to primary interference, whereas for general interference, the delay bound becomes $O\left(\#hops \cdot D^2\right)$, where D denotes the maximum degree in the conflict graph (which could be high). Jayachandran and Andrews [22] design a different scheduling scheme that ensures per-session end-to-end delay of $O\left(\#hops \cdot n\right)$. Le et al. [27] prove that max-weight scheduling has a network-average delay bound of $O\left(\#hops_{max} \cdot \theta_{max} \cdot n \cdot m\right)$. In this paper, we develop an algorithm for DCTM that improves on these bounds for a general interference model.

Our contributions.

1. *Approximation hardness of Delay-Constrained Throughput Maximization (*DCTM*)*. We show lower bounds on the computational complexity of the DCTM problem. When the wireless network is modeled as a unit-disk graph, we show that there is a constant K such that it is NP-Complete to approximate the DCTM problem within a factor of K; for arbitrary graphs DCTM is hard to approximate within a factor of $\Omega\left(n^{\epsilon}\right)$, for a constant $\epsilon \in (0,1)$, while satisfying all delay constraints. These results motivate the need of bi-criteria algorithms that violate delay bounds by some factor, in order to maximize the throughput.

2. *Multi-comodity framework for* DCTM. We develop a multi-commodity flow framework to compute a rate vector $\lambda()$ and a random-access scheduling scheme such that (i) the total throughput capacity $\sum_c \lambda(c)$ is within a factor of $O\left(\frac{\log \Delta_m}{\log\log \Delta_m}\right)$ of the maximum possible (with the given delay constraints), where $\Delta_m = \max_c\{\Delta(c)\}$, and (ii) 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 bounds include queuing delays at all intermediate nodes. In contrast to the results of [21], [22], [27], which depend on the network size or interference degrees, our

results provide per-session delay bounds, which depend only on target delays and path lengths; these are likely to be smaller, in most practical situations.

Our result involves two basic steps: (i) We derive upper-bounds on the end-to-end delays in terms of the lengths of the paths associated with the flows — our main contribution is a reduction to a simpler queuing system whose delay can be bounded by Lyapunov analysis — this reduction crucially uses the properties of random-access scheduling. (ii) An LP-rounding based approximation algorithm is devised 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 Lovasz Local Lemma [28], combined with filtering and refining steps in order to reduce certain kinds of dependencies. Our specific rounding scheme is crucial in ensuring that the loss in throughput is only $O\left(\frac{\log \Delta_m}{\log\log \Delta_m}\right)$; in contrast, a simpler direct application of randomized rounding [32] can only lead to an $O(\log n)$ factor.

3. *Quantifying the impact of adaptive channel switching*. As a specific application, we show how to estimate throughput capacity in networks with adaptive channels (e.g., in cognitive networks) and end-to-end delay requirements. These constraints can be explicitly incorporated into our framework, thereby allowing us to provably quantify the tradeoffs between these constraints. Most papers (e.g., [12], [26]) dealing with these aspects have only considered individual constraints, such as the end-to-end delay or number of channels; our approach allows all of them to be incorporated.

4. *Simulation results*. We study the empirical performance of our algorithm on small networks, and compute explicit throughput-delay tradeoffs and the saturation throughput for given delay bounds. For multi-channel networks we observe that there is significant tradeoff between the number of channels, 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 find our scheduling performs much better than our analytical delay bounds.

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 tradeoffs in any specific network, which can be useful in choosing suitable rate vectors, e.g., as in the video streaming application of [25].

Organization. We discuss related work in Section II and the network model and relevant definitions in Section III. We discuss 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 multi-channel networks in Section VII. Finally, we describe our experimental results in Section VIII and conclude in Section IX. Many proofs and some of the experimental results are omitted because of space limitations.

II. Related Work

Wired Networks: A large class of papers [36], [13], [3], [29], [30] provide analytical guarantees on the end-to-end delays and network utilization achievable through specific scheduling protocols in multi-hop *wired* networks. However, none of them 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 [17], [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., [34], [12], [42], [39], [40]. 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 has received comparatively lesser focus [21], [22], [27]. Besides these end-to-end delay results we mention in the previous section, there are also important delay results for *single-hop* traffic [33], [23], [16]. Neely [33] shows that general maximal matching policies achieve $O(1)$ network-average delay for given traffic. Kar, Luo and Sarkar [23] show that the maximum expected delay depends linearly on the chromatic number of the interference graph. However, the one-hop traffic model is crucial for these results. Gupta and Shroff [16] 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. Furthermore, our *unique* DCTM framework maximizes throughput with low per-session delay guarantees that only depend on target delays. There has also been a lot of work on random-access policies, such as [7], [41], [45], [38], but it is not clear to what extent these results can be applied for the multi-hop dynamics in arbitrary networks with general arrival processes.

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. Let $\mathcal{I}(l)$ denote the interference set containing all links that interfere with l. Our derivation of the delay bounds applies to general interference models. For results from Section VI, we assume a unit-disk model [37], 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 $d(u, v) \leq 1$. Two links interfere when one end of a link is within h hops from one end of the other, where h is a constant integer.

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.

Type	In Paper	Delay Bound	Scheduling Scheme	Efficiency Ratio	Type of Bound
End-to-end	[21]	$O\left(\#hops \cdot D^2\right)$	preemptive LIFO & stable marriage	$\Omega\left(1/D^2\right)$	per-session, upper
	[22]	$O\left(\#hops \cdot n\right)$	coordinated EDF or max-weight	$\Theta(1)$	per-session, upper
	[27]	$O\left(\#hops_{max} \cdot \theta_{max} \cdot n \cdot m\right)$	max-weight	$\Theta(1)$	network-average, upper
	Ours	$O\left((\#hops)^2\right)$	random access	$\Omega\left(1/I_{max}\right)$	per-session, upper
	[16]	non-analytical	max-weight & variants	$\Theta(1)$	network-average, lower
Single-hop	[33]	$O(1)$	maximal scheduling	$\Omega\left(1/I_{max}\right)$	network-average, upper
	[23]	$O\left(C(\mathcal{N})\right)$	max-weight or randomized indep. set	$\Theta(1)$	network-average, upper

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 l	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 [33], [16]; 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, the queue Q_l on link l is divided into logical sub-queues for each flow through l. Let $Q_{l,f}$ denote the logical sub-queue for flow f on link l. We also use the same notation to denote the backlogs. $\forall t, Q_l(t) = \sum_{f \in \mathcal{F}(l)} Q_{l,f}(t)$, 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 f's arrival packets 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 the arrival to a queue is counted in the backlog at the end of the current slot. The queue evolution mechanism can be expressed as

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

where $a_{i_f,f}(t) = \begin{cases} A_f(t), i = 1; \\ d_{(i-1)_f,f}(t), i = 2, 3, \ldots, k. \end{cases}$

B. Throughput region & 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; let OPT denote the maximum total throughput $\sum_c \lambda(c)$ for $\lambda() \in \Lambda^{OPT}$. It is known that max-weight scheduling gains the maximum throughput with traffic interior to Λ^{OPT} [14].

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 $G = (\mathcal{V}, \mathcal{L})$, a set $\mathcal{C}$ of connections, with a target delay $\Delta(c)$ for each connection c, the goal of the *Delay-constrained Throughput Maximization* (DCTM) problem is to find a stable rate vector $\lambda()$ that maximizes the total achievable rate $\sum_c \lambda(c)$, while ensuring that the session c per-packet delay is at most $\Delta(c)$. Let $OPT\left(\Delta()\right)$ denote the maximal total rate $\sum_c \lambda(c)$ that is feasible under these (delay) constraints. Note that this does not specify any scheduling scheme, and the optimum could use any scheduling scheme.

As discussed earlier, this problem is very hard, in general, and our focus is on approximation algorithms. In particular, we develop polynomial time bi-criteria approximation algorithms; we say an algorithm gives a (β_1, β_2)-approximation, if the total throughput rate guaranteed is at least $\beta_1 OPT\left(\Delta()\right)$, while the delays are at most $\beta_2 \Delta(c)$, for each session. Note that these are worst case approximation guarantees, which hold for every problem instance. In this paper, we study two kinds of bounds on the delay - average delay and the maximum delay, over all the packets for a given flow f.

[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. $\mathcal{I}(l), \mathcal{L}(f)$.

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 $p(l)$ (known as *channel access probability*) when $Q_l > 0$; 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; otherwise, the packet will stay in the queue for the next transmission service. We focus on *synchronous* random-access scheduling, where all slots are of the same length. The channel access probabilities we use are similar to [9]:

$$p(l) = 1 - e^{-e\sum_{f:\, l\in\mathcal{L}(f)} \lambda(f)/(1-\delta)}, \quad (1)$$

where $\delta \in (0,1)$ 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 gets serviced with probability $p(l,f) = \dfrac{\lambda(f)}{\sum_{f':\, l\in\mathcal{L}(f')}\lambda(f')}$.

It follows from [9] that the above random-access scheduling scheme is stable (proved in Section V) if

$$\sum_{l'\in\mathcal{I}(l)\cup\{l\}} \sum_{f:\, l'\in\mathcal{L}(f)} \lambda(f) \le \frac{1-\delta}{e}, \forall l \in \mathcal{L}. \quad (2)$$

The above constraints in Inequality (2) with RHS scaled up by eI_{max} are necessary conditions for *any* stable scheduling scheme [35], where I_{max} (also known as *maximum interference degree*) is the maximum number of links that can transmit simultaneously within any $\mathcal{I}(l)$. Inequality (2) defines a throughput region within a factor of $1/(eI_{max})$ of Λ^{OPT}. The merit of using this random-access scheduling scheme is that it achieves a throughput region comparable to that of maximal scheduling, while maintaining a low-complexity distributed manner of operation.

IV. Approximation hardness of DCTM

The DCTM is NP-complete - this follows from the fact that even without any delay constraints, the throughput maximization problem is NP-complete [43]. We extend this to show that several cases of the DTCM problem are hard 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 a gap-preserving reduction from coloring in cubic planar graphs, following the result of Clark et al. [10]; this shows that given a unit-disk graph G, it is NP-complete to distinguish between the case $\chi(G) = 3$ and $\chi(G) = 4$. 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 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 $\mathcal{L}' = \mathcal{L} \cup \{(v, v') : v \in \mathcal{V}, v'' \in \mathcal{V}''\}$. Let there be n connections (v, v') for each node $v \in \mathcal{V}$. The target delay $\Delta(c)$ for each connection c is 3. We assume that the traffic is at a constant bit-rate for each connection. 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 connections can be colored using 3 colors (this follows from a simple analysis of the reduction of [10]). Since $\Delta(c) = 3$ for each connection, at most $K'n$ connections can be scheduled within a window of size 3, implying a total throughput of at most $K'n/3$. ■

For arbitrary interference graphs, DCTM cannot be approximated within a factor of $\Omega(n^{\epsilon})$, $\epsilon \in (0,1)$; we omit the proof.

V. Delay Upper-bounds

We now derive end-to-end delay bounds for flows with a given feasible average rate vector $\lambda()$ which satisfies constraints in inequality (2).

Theorem 1: For a stable rate vector $\lambda()$ that satisfies inequality (2), the random-access scheduling protocol defined by (1) ensures that (i) the average delay for each flow f is $O\left(|\mathcal{L}(f)|^2/(\lambda(f))^2\right)$; and (ii) the average network delay is $O\Big(\sum_{f\in\mathcal{F}}|\mathcal{L}(f)|^2 / \big(\sum_{f\in\mathcal{F}}\lambda(f)\min_{f\in\mathcal{F}}\{\lambda(f)\}\big)\Big)$.

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 $x(l,f) = \lambda(f)/(1-\delta)$, and $x(l) = \sum_{f\in\mathcal{F}(l)} x(l,f)$. Then equation (1) can be rewritten as $p(l) = 1 - e^{-ex(l)}$, and we have:

$$\begin{aligned}\mathbb{E}\{\mu_{l,f}(t)\} &\ge p(l,f)p(l)\textstyle\prod_{l'\in\mathcal{I}(l)}(1-p(l'))\\ &\ge p(l,f)\left(1-e^{-ex(l)}\right)e^{ex(l)-1}\\ &= \frac{\left(e^{ex(l)}-1\right)x(l,f)}{ex(l)} \ge x(l,f).\end{aligned} \quad (3)$$

The idea for the proof of the above theorem is that due to the properties of random-access scheduling, each flow can be viewed in "isolation" as a tandem system, with expected lower bounds on the service rates $\mu(l,f)$ that only depend on $x(l,f)$ for each logical queue $Q_{l,f}$, as shown in Equation (3). Let the triple $(Q(), a(), \mu())$ denote a queueing system. From now on till the end of this section, we use R to denote the basic queueing system under the basic scheduling scheme mentioned in Section III, and 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 omitted because of the space limitation.

Lemma 2: Let R and R_1 be two tandem queueing systems with the same arrival processes but $\mathbb{E}\left\{\mu^R_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\mu^{R_1}_{i_f,f}(t)\right\}$, for each flow f and link i_f, at t. Then,

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

(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 subject to a Bernoulli distribution with $\mathbb{E}\left\{\mu^{R_1}_{i_f,f}(t)\right\} = \lambda(f) + \frac{i\epsilon(f)}{|\mathcal{L}(f)|} \leq x^R(i_f, f) \leq \mathbb{E}\left\{\mu^{R}_{i_f,f}(t)\right\}$, where $\epsilon(f) = \frac{\delta\lambda(f)}{1-\delta}$. Then, Lemma 2 implies $\mathbb{E}\left\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R_1}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R}_{i_f,f}(t)\right\}$. Note that whether the reduced system is using wireless medium no longer matters.

(2) **Reduction 2**. Note that the exogenous arrival at the source link of the tandem system $(Q^{R_1}(), a^{R_1}(), \mu^{R_1}())$ is a general arrival process, making it nontrivial and hard to use earlier methods directly, e.g., [6], [11], [20], in bounding the end-to-end delays. Therefore, we reduce the system R_1 to another queueing system R_2 so that the arrival process for each non-source queue is 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 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 limited, 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 arrival $\mathbb{E}\left\{a^{R_2}_{i_f,f}(t)\right\} = \mathbb{E}\left\{\mu^{R_1}\left((i-1)_f, f\right)(t)\right\}$ is also a Bernoulli process. This leads us to the following lemma.

Lemma 3: At every time slot t, $\mathbb{E}\left\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R_2}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R_1}_{i_f,f}(t)\right\} \geq \mathbb{E}\left\{\sum_{i=1}^{|\mathcal{L}(f)|} Q^{R}_{i_f,f}(t)\right\}$.

The proof can be carried out by induction on t.

(3) **Queueing analysis for** R_2. The fact that the arrival and service process of each $Q^{R_2}_{i_f,f}$ is subject to geometric distribution, allows us to perform an isolated queueing analysis for each $Q^{R_2}_{i_f,f}$. For any link $i = 1, 2, \ldots, |\mathcal{L}(f)|$, we have

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

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], [33] for the details of this approach. We define the Lyapunov function as $L\left(Q^{R_2}_{i_f,f}(t)\right) \triangleq \left(Q^{R_2}_{i_f,f}(t)\right)^2$.

The 1-step Lyapunov drift is then defined as below:

$$\begin{aligned}\triangle^{(1)}_Q\left(Q^{R_2}_{i_f,f}(t)\right) &\triangleq \mathbb{E}\left\{L\left(Q^{R_2}_{i_f,f}(t+1)\right) - L\left(Q^{R_2}_{i_f,f}(t)\right)\right\}\\ &\leq \mathbb{E}\left\{\left(\mu^{R_2}_{i_f,f}(t)\right)^2 + \left(a^{R_2}_{i_f,f}(t)\right)^2\right\} -\\ &\quad \mathbb{E}\left\{2Q^{R_2}_{i_f,f}(t)\left(\mu^{R_2}_{i_f,f}(t) - a^{R_2}_{i_f,f}(t)\right)\right\}.\end{aligned} \quad (4)$$

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 $\mathbb{E}\left\{\mu^{R_2}_{i_f,f}(t) - a^{R_2}_{i_f,f}(t)\right\} = \epsilon(f)/|\mathcal{L}(f)|$. 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 (4) can be then rewritten as

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

From Theorem 1 in [33], the average backlog $\overline{Q}^{R_2}_{i_f,f}$ is $\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{Q^{R_2}_{i_f,f}(\tau)\right\} \leq \frac{1+\max\left\{1, eA^{(2)}\right\}}{2e\epsilon(f)}|\mathcal{L}(f)|$. Therefore, the sum of backlogs for f is

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

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

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

Theorem 1 now follows.

VI. Multi-commodity Flows with Delay Guarantees

From Theorem 1, it follows that the average end-to-end delay bound for flow f is quadratically in proportion to $|\mathcal{L}(f)|$, and inversely quadratically in proportion to $\lambda(f)$; the average network average end-to-end delay bound is in proportion to $\sum_{f\in\mathcal{F}} |\mathcal{L}(f)|^2$ and $\sum_{f\in\mathcal{F}} 1/\lambda(f)$, and inversely in proportion to $\sum_{f\in\mathcal{F}} \lambda(f)$. 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., keep $\lambda(f)$ high) and "short" paths (i.e., 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 vector $\lambda()$ such that: (i) the total rate $\sum_c \lambda(c)$ is "close" to $OPT(\Delta())$; more precisely, we have $\sum_c \lambda(c) = \Omega\left(\frac{\log\log\Delta_m}{\log\Delta_m}\right) OPT(\Delta())$ (ii) for each c, each flow f for connection c if $\lambda(f) > 0$, we have $\lambda(f) = \frac{\log\log\Delta_m}{\log\Delta_m}$ (iii) for each c, each flow path f for connection c with $\lambda(f) > 0$ has length at most $2\Delta(c)$. Using rate vector $\lambda()$ along with the random-access scheduling scheme described in Section V leads to a (β_1, β_2) bi-criteria approximation, where the total throughput is at least $\beta_1 OPT$ and the delays are $\beta_2\Delta(c)$ for connection c.

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. Our approach employs an approximation algorithm, which selectively drops some cost-unfavorable sessions and maximizes the rates of the rest sessions. Path constrained flows have been studied in a wired setting, e.g., [5], [18], but the interference constraints, and the fact that we need both short paths and large flow values make our problem different. Our algorithm involves the following steps.

(Step 1) **Linear programming formulation**. 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, but in most of this paper, the cost of a path will denote its length in hops. 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 (5a) to (5e). Here, $y(c)$ denotes the total rate for connection c, and $y(f)$ the rate along path $f \in \mathcal{F}(c)$; $y(l, c) = \sum_{f\in\mathcal{F}(c):l\in\mathcal{L}(f)} y(f)$ is the total flow for c along l.

LP: $\max \sum_c y(c)$

$$\text{s.t.}\quad \forall c,\ y(c) = \sum_{f\in\mathcal{F}(c)} y(f) \tag{5a}$$

$$\forall c,\ \sum_{f\in\mathcal{F}(c)} y(f)cost(f) \le \Delta(c)y(c) \tag{5b}$$

$$\forall l, c,\ y(l, c) = \sum_{f\in\mathcal{F}(c):\ l\in\mathcal{L}(f)} y(f) \tag{5c}$$

$$\forall l,\ \sum_{l'\in(\mathcal{I}\cup\{l\})} \sum_c y(l', c) \le \frac{1-\delta}{e} \tag{5d}$$

$$\forall f,\ y(f) \ge 0 \tag{5e}$$

In the above formulation, constraints (5a) and (5c) represent path-based flow-conservation constraints; (5b) constrains total path cost which in our case is the path length, since end-to-end delay is lower bounded by the number of hops; congestion constraints in (5d) ensure the stability under a random-access scheme. 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 constraint (5a) and (5c) with $\forall c$, $\sum_{l\in\mathcal{L}_{out}(s(c))} y(l, c) = y(c)$ and $\sum_{l\in\mathcal{L}_{in}(t(c))} y(l, c) = y(c)$, and flow-conservation constraints at all other nodes;

(2) replacing constraint (5b) with $\forall c$, $\sum_{l\in\mathcal{L}} y(l, c)cost(l) \le \Delta(c)y(c)$; and

(3) replacing constraint (5e) with $y(l, c) \ge 0, \forall(l, c)$.

Let $y^*()$ denote the optimum fractional solution to the above LP. 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**. We transform $y^*()$ into another fractional solution $y'()$ in the following manner:

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

It follows by a simple averaging argument, that $\forall c,\ y'(c) = \sum_{f\in\mathcal{F}^*(c,2\Delta(c))} y'(f) \ge y^*(c)/2$.

(Step 3) **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.

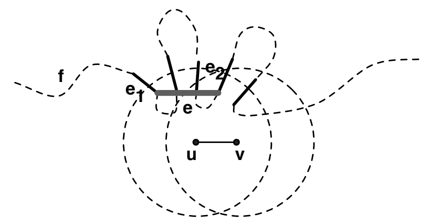


Fig. 1: Path refinement operation: Consider path f (shown by dashed curved line) and link $l = (u, v)$. Path f revisits $\mathcal{I}(l)$ multiple times, and the segment of f from between the endpoints of edges e_1 and e_2 can be replaced by edge e (shown in light gray and bold) to get a path f' which is shorter, and sending the same flow on f' (instead of f) is still feasible.

(Step 4) **Partitioning into cells**. Let γ denote the number of congestion constraints (as in Constraint (5d)) that involve a given path $f \in \mathcal{F}^*$ from the previous step. It is crucial that γ be "small", so that the rounding scheme in later steps produces a "good" approximate ratio. However, γ becomes $O(\Delta_m \max_l |\mathcal{I}(l)|)$ if we use the original set of congestion constraints in Constraints (5d), which is the case for general interference model. To control this, we now coarsen the formulation in the following manner: (i) γ is upper-bounded by $O(\Delta_m)$; and (ii) the coefficient of any flow rate variable in the new congestion constraints is in $[0, 1)$. It is only in Steps 3 and 4 that we require a unit-disk graph model described in Section III. The construction involves the following sub-steps:

(1) We partition the plane into $\frac{1}{2} \times \frac{1}{2}$ grid cells. Let $\mathcal{B}$ denote the set of cells and let $b \in \mathcal{B}$ denote a cell; we say a link $l \in b$ if and only if one end of l falls in b. The number of cells that a path $f \in \mathcal{F}^*$ goes through is hence $O(\Delta_m)$.

(2) Constraints (5d) imply the following:

$$\forall b \in \mathcal{B},\ \sum_{l\in b} \sum_{f:l\in\mathcal{L}(f)} y(f) \le \tfrac{1-\delta}{e}. \tag{6}$$

(3) We then divide both sides of Inequality (6) by a constant factor of K_0 (mentioned in Step 3), which is the maximum number of links in any interference set that belongs to the same path, such that the coefficient of any rate variable becomes less than 1.

(Step 5) **Rounding and selection**. We now select a set of paths $\mathcal{F}' \subseteq \mathcal{F}^*$ for the connections, so that: (i) we choose at most one path for each session c; (ii) the rate on c is $z(c) = 1$; (iii) $|\mathcal{F}'| = \sum_c z(c)$ is large enough (in our algorithm, we ensure that this is $\Theta(\sum_c y'(c))$); and (iv) the chosen paths incur "low" congestion; more precisely, for each link l, we have $\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_c z(l', c) \leq K_1 \log\log \Delta_m / \log \Delta_m$, where $z(l', c) \in \{0, 1\}$ is the rate on path $f \in \mathcal{F}' \cap \mathcal{F}(c)$ such that $l' \in \mathcal{L}(f)$, and K_1 is a constant. We employ a randomized rounding technique based on [28], and this involves the following sub-steps:

(1) We partition $\mathcal{C}$ into groups $\mathcal{C}_1, \ldots, \mathcal{C}_k$ such that for each i (for all but possibly one group), $\sum_{c \in \mathcal{C}_i} y'(c) \in [1, 2)$; w.l.o.g., if there is a group i with $\sum_{c \in \mathcal{C}_i} y'(c) < 1$, we assume that $i = k$.

(2) We construct a new fractional vector $y''()$ in the following manner: for each i if $\sum_{c \in \mathcal{C}_i} y'(c) = a > 1$, for all $c \in \mathcal{C}_i$, and for all $f \in \mathcal{F}(c)$, we define $y''(f) = y'(f)/a$.

(3) Now apply the rounding algorithm of [28] to choose a path f_i for each group $\mathcal{C}_i$ (except group $\mathcal{C}_k$, in case $\sum_{c \in \mathcal{C}_k} y'(c) < 1$), using the congestion constraints constructed in Step 4, instead of (5d). Let $\mathcal{F}' = \{f_1, \ldots, f_k\}$ be the paths which are chosen.

(Step 6) **Scaling and choosing flow vector**. We choose a rate vector $\lambda()$ so that: $\lambda(f_i) = K_2 \log\log \Delta_m / \log \Delta_m, \forall f_i \in \mathcal{F}'$, where K_2 is a constant. Note that for some connections, no flows would be chosen; for each ("original") connection c, define $\lambda(c)$ as total flow of c; $\lambda(c) = \lambda(f_i)$ if $f_i \in \mathcal{F}(c)$.

Lemma 4: Let $\lambda()$ denote the rate vector computed by the additional path transformations. Then, we have: (i) $\sum_c \lambda(c) = \Omega\left(\frac{\log\log \Delta_m}{\log \Delta_m}\right) \sum_c y^*(c)$, where $y^*()$ is the optimum fractional solution to (LP), (ii) for each connection c, on the path f chosen for this connection with $\lambda(f) > 0$, we have $cost(f) \leq 2\Delta(c)$, and (iii) $\lambda()$ is a feasible solution to (LP), and, in particular, satisfies the stability constraints needed for our random-access scheduling protocol.

Proof: (sketch) As mentioned earlier, after the filtering step, for each c, we have $y'(c) \geq y^*(c)/2$, and all paths f with $y'(f) > 0$ have $cost(f) \leq 2\Delta(c)$. Step 3 only alters the links on the paths, but the flow rates remain unchanged. Step 4 only constructs a new auxiliary set of constraints, and does not change any of these quantities. Also, the path lengths do not increase in Steps 3, 4. Therefore, property (ii) above holds.

Next, after sub-step 1 of Step 5, note that $\sum_i \sum_{c \in \mathcal{C}_i} y'(c)$ remains unchanged. For the rest of this proof, we assume that the number of connections $|\mathcal{C}|$ is at least a constant K_3 (if $|\mathcal{C}|$ is smaller, the argument below can be easily modified), so that $\sum_c y^*(c)$ can be assumed to be larger than a constant K_4. After the scaling in substep 2 of Step 5, we have $\sum_{i<k} \sum_{c \in \mathcal{C}_i} y''(c) \geq \sum_c y'(c)/2 \geq \sum_c y^*(c)/4$, because the flow in each group $\mathcal{C}_i$ is reduced by at most a factor of 2, relative to the vector $y'()$. Also, observe that $y''()$ is a feasible solution for (LP), since for each f, we have $y''(f) \leq y^*(f)$.

Finally, we can apply the rounding result of [28] in substep 3 of Step 5. For each group $\mathcal{C}_i$, $i < k$, we have $\sum_{c \in \mathcal{C}_i} \sum_{f \in \mathcal{F}'(c)} y''(f) = 1$. By adding a super-source s'_i and super-sink t'_i, we can view these paths to be between s'_i and t'_i in a modified graph. In the rounding process, we only consider the congestion induced on the cells in $\mathcal{B}$ (instead of directly on each interference set). The crucial aspect is that the number of cells whose congestion constraint involve a given path variable $y(f)$ is bounded by $O(\Delta_m)$. Therefore, by applying the randomized rounding approach of [28] based on the constructive version of the Local Lemma, we have the following properties: (i) for each $\mathcal{C}_i$, exactly one path f_i with rate of 1 is chosen; and (ii) for each cell b, we have $\sum_{l \in b} \sum_{f: l \in \mathcal{L}(f)} z(f) = O\left(\frac{\log \Delta_m}{\log\log \Delta_m}\right)$, where $z(f)$ is the rate of path f after rounding. Since only one path is chosen in each $\mathcal{C}_i$, each connection c has at most one path. Further, because any interference set can be covered by $O(h^2)$ (which is a constant) cells, we have $\sum_{l' \in \mathcal{I}(l) \cup \{l\}} \sum_c z(l', c) = O\left(\frac{\log \Delta_m}{\log\log \Delta_m}\right)$, where $z(l', c) \in \{0, 1\}$ is the rate of connection c on l' after rounding. Note that though the algorithm of [28] is randomized, the above properties hold with probability 1. In Step 6, we scale down the flow rates by a factor of $O\left(\frac{\log \Delta_m}{\log\log \Delta_m}\right)$, and thus $\lambda(f_i) = K_2 \sum_{c \in \mathcal{C}_i} \frac{\log\log \Delta_m}{\log \Delta_m} y''(c)$ for each such path f_i, for a constant K_2. Therefore, $\sum_c \lambda(c) = \sum_i \lambda(f_i) = \sum_i \sum_{c \in \mathcal{C}_i} \frac{\log\log \Delta_m}{\log \Delta_m} y''(c) K_2 = \Omega\left(\frac{\log\log \Delta_m}{\log \Delta_m}\right) \sum_c y^*(c)$, implying statement (i) of the lemma. Finally, since $\lambda(l', c) = \frac{\log\log \Delta_m}{\log \Delta_m} z(l', c) K_2$ for each l', c, statement (iii) follows from the above property of $z()$'s. ■

Combining the delay analysis in Section V, gives us the following bi-criteria approximation result for DCTM problem.

Theorem 2: Let $\lambda()$ be the rate vectors resulted from above. Then, we have $\sum_c \lambda(c) = \Omega\left(\frac{\log\log \Delta_m}{\log \Delta_m}\right) OPT(\Delta())$ and $\lambda(f) = \Omega\left(\frac{\log\log \Delta_m}{\log \Delta_m}\right)$. The random-access scheduling protocol ensures that, for each session c and each flow $f \in \mathcal{F}(c)$, the average delay is $O\left(\left(\frac{\log \Delta_m}{\log\log \Delta_m} \Delta(c)\right)^2\right)$.

VII. Quantifying the Delays from Adaptive Channel Switching

While adaptive channel switching capabilities in recent MC-MR and cognitive radio devices have led to throughput improvements [31], [24], non-negligible switching delays may be incurred. We show how our formulation can incorporate these delays to study multi-channel systems.

Here, we discuss a single radio interface per node; this can be easily extended to the case of multiple interfaces. Let Ψ denote the set of channels available in the system; let ψ, ψ' be two arbitrary channels in Ψ. If l and l' are incoming and outgoing links of a node respectively, let the delay in switching from channel ψ to ψ' be denoted by $d(\psi, \psi')$. Our formulation in Section VI is based on link delays, whereas switching delays are not captured because they are associated with nodes. The difficulty of applying the LP formulation lies in adapting Constraints (5c) and (5d) to multi-channel model. We tackle this by performing a graph transformation on the network graph G to a new graph G' as follows. (1) We *split* each link in G into $|\Psi|$ links, each associated with a unique channel; (2) for each node $v \in G$, we *split* it into $\big(|\mathcal{L}_{in}(v)| + |\mathcal{L}_{out}(v)|\big)|\Psi|$

nodes, each of which is incident with only one incoming or outgoing link. (3) each node incident with an incoming link is connected to each node incident with an outgoing link, by an intermediate link associated with a switch delay.

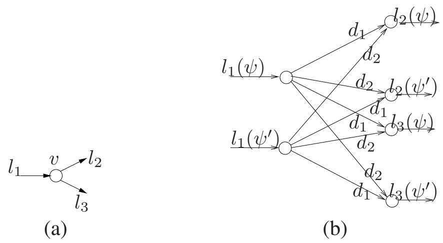


Fig. 2: (a) Node v with incoming link l_1, outgoing links l_2, l_3, and channels ψ, ψ'. (b) The reduction after node and link splitting with addition of switching link with delays d_1, d_2.

Let $l(\psi)$ denote the link associated with channel ψ in G' emerged from link l in G, and let $v(\psi, \psi')$ denote the intermediate link switching from channel ψ to ψ in G' emerged from node v in G. Let $\mathcal{L}'$ denote the set of links in G', and let $\mathcal{L}'_{(1)}(v)$ and $\mathcal{L}'_{(2)}(v)$ denote the sets of new links emerging from step (1) and (2) above respectively for node v. Figure 2 shows an example of transforming the original network graph in Figure 2a to the graph in Figure 2b, where switch delays are $d_1 = d(\psi, \psi) = d(\psi', \psi')$ and $d_2 = d(\psi, \psi') = d(\psi', \psi)$.

For $l \in G$, let $Pri(l)$ denote the primary interference set which includes all links in G sharing an end node with l. After graph transformation, for link $l(\psi)$, let $\lambda(l(\psi)) = \sum_{c \in \mathcal{C}} \lambda(l(\psi), c)$. The stability condition [8] can be written as

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

In the above inequality, note that l and l' denote links from G; the link-channel pair $l(\psi)$ denotes a link from G'. Additionally, we construct interference constraints on the intermediate switching links in G', depending on specific switching conditions. For example, we can use

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

Then after modifying the flow conservation conditions and using switching delay as the link cost for any intermediate switching link, we are able to adapt the LP in Section VI to finding a multi-commodity flow vector for the multi-channel model. By using the distributed random-access scheduling scheme of [19], and setting $p(l, f)$ as in Section III, we can obtain results in the same form as Theorem 2.

VIII. Simulation results

We conduct simulation study and visualize the delay-throughput-channel tradeoffs by solving out the LP's from Section VI. For simplicity, we used a uniform target delay for all sessions. First, for single-channel models, we show how the optimal network throughput value reflects the variation of the uniform target delay, number of sessions and network size. Next, for multi-channel models, we show the trend of 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.

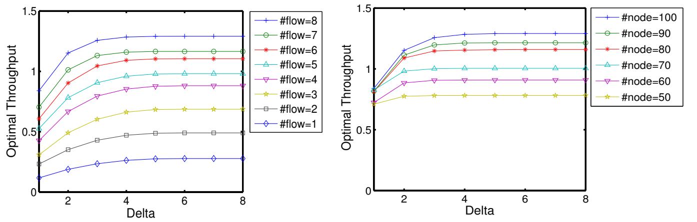
Single-channel networks. We generated random unit-disk graphs with varying sizes, and varied the number of random connections for a network topology. For each choice of network size, number of connections and Δ value, we perform 500 iterations of random topology and connection generation, plus LP formulation. Figure 3 shows numerical tradeoff curves under the same interference model. Figure 3a features a fixed network size of 100, and Figure 3b features a fixed number of flows equal to 8. Intuitively, as Δ values increase, thereby loosening the delay constraint, the optimal throughput will rise; as the number of random connections increases, the optimization process gets more exploration space, yielding greater optimal network throughput. The saturation of the curves happen where the interference plays a major role through the constraint for stability in the LP.

Multi-channel networks. Figure 4a and 4b show the optimal throughput calculated by solving the LP's for grid topologies with 2-hop interference model on grid topologies. As expected, the total throughput increases as additional channels are equipped and delay bound is loosened. Saturation points are observed in both plots. Addition of channel resources alleviates the severance of interference, thus yielding a slower saturation process. Also, loosening the delay bound produces similar effects, and the addition of channels make the optimization process to explore more of the delay bound.

IX. Conclusion and open problems

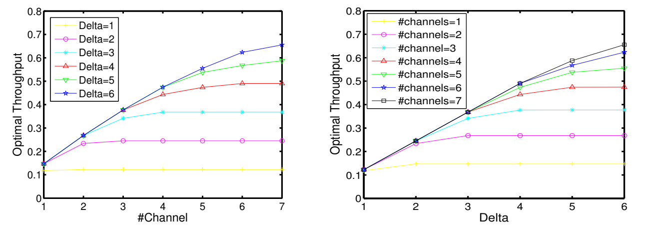
Characterizing delay-throughput tradeoffs and bounds is a fundamental problem in wireless networks, with numerous applications. In this paper, we develop a theoretical framework to rigorously bound this tradeoff and provably approximate the maximum throughput with given per-session delay requirements. Extending these techniques to bound the average per-session delay with additional fairness constraints is a very challenging open problem. Maximal scheduling [33] also fits in our DCTM framework. However, bounding end-to-end delay for maximal scheduling is an open and difficult problem. New progress on this will likely enable our DCTM framework for maximal scheduling.

Acknowledgment. We thank our external collaborators and members of the Network Dynamics and Simulation Science Laboratory (NDSSL) for their suggestions and comments. We also thank the referees. The work of the first two authors 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) Throughput-delay trend with varying #flows on 100-node topo (b) Throughput-delay trend with 8 flows and varying network sizes

Fig. 3: Trade-offs among OPT throughput, delay, number of flows, network size.



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

Fig. 4: Trade-offs among OPT throughput, delay, number of channels.

REFERENCES

[1] T. Achterberg. Scip: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):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] M. Andrews and L. Zhang. 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)*, pages 679–690, 2006.

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

[7] N. Bisnik and A. Abouzeid. Queuing network models for delay analysis of multihop wireless ad hoc networks. *Ad Hoc Networks*, 7(1), 2009.

[8] M. M. S. P. A. S. Bo Han, V. S. Anil Kumar. Distributed strategies for channel allocation and scheduling in software-defined radio networks. In *IEEE Infocom*, pages 1521–1529, 2009.

[9] 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.

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

[11] M. Draief, J. Mairesse, and N. O'Connell. Queues, stores, and tableaux. *Journal of Applied Probability*, 42(4):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.*, 1(1):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.*, 10:477–486, 2002.

[16] G. R. Gupta and N. B. Shroff. Delay analysis for multi-hop wireless networks. In *IEEE Infocom*, 2009.

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

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

[19] B. Han, V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. Distributed strategies for channel allocation and scheduling in software-defined radio networks. In *IEEE Infocom*, 2009.

[20] J. Hsu and P. Burke. Behavior of tandem buffers with geometric input and markovian output. *IEEE Trans. on Comm.*, 24(3):358–361, 1976.

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

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

[23] K. Kar, X. Luo, and S. Sarkar. Asymptotically tight delay guarantees through wireless link scheduling. In *IEEE Infocom*, 2009.

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

[25] S. Kompella, S. Mao, Y. Hou, and H. Sherali. On path selection and rate allocation for concurrent video sessions in ad hoc networks. *IEEE/ACM Trans. on Netw.*, 17(1):212–224, 2009.

[26] P. Kyasanur and N. H. Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *MobiCom*, pages 43–57, 2005.

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

[28] F. T. Leighton, C. J. Lu, S. B. Rao, and A. Srinivasan. New algorithmic aspects of the local lemma with applications to routing and partitioning. *SIAM J. Comput.*, 31:626–641, 2001.

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

[30] C. Li and E. Knightly. Schedulability criterion and performance analysis of coordinated schedulers. In *ITC*, 2001.

[31] X. Lin and S. Rasool. A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad-hoc wireless networks. In *IEEE Infocom*, pages 1118–1126, 2007.

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

[33] M. J. Neely. Delay analysis for maximal scheduling with flow control in wireless networks with bursty traffic. *IEEE/ACM Trans. on Netw.*, 17(4):1146–1159, 2009.

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

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

[36] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Trans. on Netw.*, 2(2), 1994.

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

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

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

[40] 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.*, 15(5):981–992, 2007.

[41] 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.

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

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

[44] R. Wunderling. *Paralleler und objektorientierter Simplex-Algorithmus*. PhD thesis, Technische Universität Berlin, 1996.

[45] M. Xie and M. Haenggi. Towards an end-to-end delay analysis of wireless multihop networks. *Ad Hoc Networks*, 7(5):849–861, 2009.