Scheduling for Optimal Rate Allocation in Ad Hoc Networks With Heterogeneous Delay Constraints

Juan José Jaramillo, *Member, IEEE,* R. Srikant, *Fellow, IEEE,* and Lei Ying, *Member, IEEE*

***Abstract*—This paper studies the problem of scheduling in single-hop wireless networks with real-time traffic, where every packet arrival has an associated deadline and a minimum fraction of packets must be transmitted before the end of the deadline. Using optimization and stochastic network theory we study the problem of scheduling to meet quality of service (QoS) requirements under heterogeneous delay constraints and time-varying channel conditions. Our analysis results in an optimal scheduling algorithm which fairly allocates data rates to all flows while meeting long-term delay demands. We also prove that under a simplified scenario our solution translates into a greedy strategy that makes optimal decisions with low complexity.**

***Index Terms*—Wireless networks, ad hoc networks, quality of service, scheduling, real-time traffic.**

I. Introduction

In this paper we study the problem of scheduling real-time traffic in ad hoc networks under maximum per-packet delay constraints. The problem of scheduling best-effort traffic, which is defined as traffic that does not have any kind of quality of service (QoS) requirements such as minimum bandwidth or maximum delay, has been extensively studied for the case of wireless networks. An optimization framework for resource allocation in wireless networks has been developed in [1]–[6], where a dual decomposition approach was used to derive various components of the resource allocation architecture such as scheduling, congestion control, routing, power control, etc. A striking feature of the solution is an alternative derivation of the maxweight algorithm proposed in [7]. We refer the readers to [8], [9] for a survey of these works.

Scheduling algorithms for packets with strict deadline requirements have been proposed in [10]–[13], but the solutions are only approximate. In [14]–[16], the problem of optimal admission control and scheduling for real-time traffic was addressed for access-point wireless networks in which only one link can transmit at any given time. Among the many contributions in these papers is a key modeling innovation whereby the network is studied in frames, where a frame is a contiguous set of time-slots of fixed duration. Packets with deadlines are assumed to arrive only at the beginning of a frame and have to be served before the end of the frame according to some specified deadlines.

The problem of optimal congestion control and scheduling for general ad hoc networks and arrivals was studied in [17], using the modeling paradigm of frames proposed in [14]. The model allows a common framework for handling both best-effort and real-time traffic simultaneously, but it can only handle homogeneous per-packet delay requirements.

In this paper, we further extend the results of [17] for the case of heterogeneous delays and time-varying channel conditions.

The main contributions of this work are summarized as follows:

1) We show that the framework in [17] can be extended to heterogeneous delays for the case of periodic traffic, which is an important practical extension since many examples of real-time traffic fall into this category.
2) We then consider noisy channels where the transmitter does not have perfect channel state information and relies on feedback from the receiver to find out if a transmission was successful. The case with per-slot feedback is a non-trivial practical and theoretical extension. The difficulty arises due to per-slot feedback because the scheduler's decision at each time instant is a policy (i.e., a mapping from observed feedback so far in the frame to a scheduling decision). We have shown that the scheduling decisions in this case can be solved using a dynamic program encountered at the beginning of each frame. The dynamic program formulation provides a systematic way to solve for the optimal solution.
3) The usefulness of the dynamic program solution is further demonstrated in the case of collocated networks, where only one link can transmit at any given time. For this case, we are able to provide a simple proof that a greedy solution is optimal. The result shows the derivative nature of the optimization-based approach even in the per-slot feedback case: the solution can be simply derived using the optimization-decomposition cookbook. Unlike prior solutions using the optimization-decomposition approach, we now have a dynamic program component.

The paper is organized as follows. Section II presents the network model we use in this work. The optimization formulation is presented in Section III for the simplest of the channel models we study, while the dual decomposition approach is developed in Section IV. The optimal scheduler and its convergence properties are presented in Section V. Since we study two different channel models, in Section VI we highlight the differences between the two models, and show

Research supported by NSF Grants 07-21286, 05-19691, 03-25673, 08-31756, 09-53165, ARO MURI Subcontracts, AFOSR Grant FA-9550-08-1-0432, and DTRA Grants HDTRA1-08-1-0016, HDTRA1-09-1-0055.

J. J. Jaramillo and L. Ying are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA. Email: {jjjarami, leiying}@iastate.edu

R. Srikant is with the Department of Electrical and Computer Engineering, and the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801 USA. Email: rsrikant@illinois.edu

the relationship between feedback after every transmission and algorithm complexity. In Section VII we perform a simulation study to observe the rates that can be achieved under different channel models. Finally, in Section VIII we present the conclusions.

II. Network Model

In this section we present our model for a network composed of single-hop traffic flows, such that each packet has a maximum delay constraint.

We represent the network using a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of links, such that for any $n_1, n_2 \in \mathcal{N}$, if $(n_1, n_2) \in \mathcal{L}$ then node n_1 can communicate to node n_2. Links are numbered 1 through $|\mathcal{L}|$, and by abusing notation, we will sometimes use $l \in \mathcal{L}$ to mean $l \in \{1, 2, \ldots, |\mathcal{L}|\}$.

Time is assumed to be divided in slots, and a set of T consecutive slots is called a *frame*. Let $\mathcal{T} \overset{def}{=} \{1, \ldots, T\}$. We denote by $a = (a_{lt})_{l \in \mathcal{L}, t \in \mathcal{T}}$ the number of packet arrivals at a given frame for link l at time slot t, and assume that we get to know a at the beginning of the frame[1]. Furthermore, assume that $a_l \overset{def}{=} \sum_{t \in \mathcal{T}} a_{lt}$ has mean λ_l.

Define $\mathcal{T}_l^a \overset{def}{=} \{t : t \in \mathcal{T} \text{ and } a_{lt} > 0\}$ to be the set of arrival times at link l. Let $\tau = (\tau_{lt})_{l \in \mathcal{L}, t \in \mathcal{T}_l^a}$ be the deadline associated with packet arrivals. That is, a packet that arrived at link l, time t, must be transmitted by the end of time slot τ_{lt}. Note that this assumption means that if multiple packets arrive in the same time slot for a given link, they will all have the same deadline. This assumption can be relaxed by properly changing the notation, but we avoid doing this to simplify the presentation. We also assume that the deadlines are such that

$$\{t_1, \ldots, \tau_{lt_1}\} \cap \{t_2, \ldots, \tau_{lt_2}\} = \emptyset \text{ for all } t_1, t_2 \in \mathcal{T}_l^a$$

and

$$\tau_{lt} \leq T \text{ for all } l \in \mathcal{L}, t \in \mathcal{T}_l^a.$$

In other words, packets must be transmitted before the next set of arrivals occurs in subsequent time slots, and all packets must be transmitted before the end of the frame.

If a packet misses its deadline it is discarded, and it is required that the loss probability at link $l \in \mathcal{L}$ due to deadline expiry be no more than p_l. To avoid unnecessary complexity in the formulation, we will write a to denote both the number of packet arrivals and its associated deadlines τ.

This paper studies the cases when the channel state is assumed to be constant for the duration of a frame as well as when it is allowed to change from time slot to time slot. In both cases we assume the state is independent across frames (time slots, respectively) and independent of arrivals. When the channel is fixed in a frame, let $c = (c_l)_{l \in \mathcal{L}}$ denote the number of packets link l can successfully transmit in a time slot. When the channel is allowed to change, define $c = (c_{lt})_{l \in \mathcal{L}, t \in \mathcal{T}}$ to be the number of packets that can be successfully transmitted at link l in time slot t.

TABLE I
Summary of Notation

	Meaning
$\mathcal{L}$	Set of links
T	Number of time slots in a frame
$\mathcal{T}$	Set of first T positive integers. $\mathcal{T} = \{1, \ldots, T\}$
a_{lt}	Number of packet arrivals in a given frame at link l, time slot t
a_l	Total number of arrivals in a given frame at link l.
$\mathcal{T}_l^a$	Set of arrival times at link l. $\mathcal{T}_l^a = \{t : t \in \mathcal{T} \text{ and } a_{lt} > 0\}$
τ_{lt}	Deadline associated with packet arrivals at link l, time slot t
p_l	Maximum allowable loss probability at link l for packets that missed their deadline
c_l	Number of packets link l can successfully transmit in a time slot. Random variable with mean $\bar{c}_l$ and variance σ_{cl}^2

If we get to know the channel state before transmission, we can determine the optimal rate at which we can successfully transmit, possibly allowing more than one packet to be transmitted in a single slot. On the other hand, if the channel state is not known, we can only determine whether a transmission was successful or not after we get some feedback from the receiver. In this paper we try to capture these different scenarios in the following cases:

1) Known channel state: It is assumed that c_l is a *non-negative* random variable with mean $\bar{c}_l$ and variance σ_{cl}^2, and we get to know the channel state at the *beginning* of the frame.
2) Unknown channel state, per-slot feedback: It is assumed that c_{lt} is a *Bernoulli* random variable with mean $\bar{c}_l$ and we get to know the channel state at the *end* of the time slot. In other words, acknowledgments are received after each transmission.

In the known channel state case we can potentially send more than one packet in a time slot at higher rates since channel estimation allows us to determine the optimal transmission rate. This is the reason why we make no assumptions on the values that c_l can take since it will be determined by the particular wireless technology used. In the case where the channel is unknown before transmission, we assume that we only get binary feedback in the form of acknowledgments, which is reflected in the Bernoulli assumption on c_{lt}. Thus, in this case, without any loss of generality, we assume only one packet can be transmitted per time slot per link. It must be noted that while more realistic channel models can be used in the formulation, the solution to the problem does not significantly change from the one presented in this paper. We have chosen this simple model for the sake of simplicity in the exposition.

For convenience, the definitions used are summarized in Table I. For the sake of simplicity in the presentation, we will first develop the known channel case and we will later highlight the differences between this case and the other case in Section VI.

[1] In practice, this assumption is valid for periodic traffic, as is the case for voice and video applications. The general treatment that we develop allows for the case of coders that have periodic traffic of variable length. One example of this is the well-known MPEG-2 standard [18], which defines *a priori* a group of pictures (GOP) structure that specifies the order in which intra- and inter-frames are arranged in a periodic sequence.

III. Static Problem Formulation

To design our algorithm, we will first formulate the problem as a static optimization problem. Using duality theory, we will then obtain a dynamic solution to this problem and later we will prove its properties using stochastic Lyapunov techniques.

Let $s = (s_{lt})_{l\in\mathcal{L},t\in\mathcal{T}}$ denote the number of packets scheduled for transmission at link l and time slot t. We will only focus on feasible schedules, so if $s_{l_1t} > 0$ and $s_{l_2t} > 0$ for any t, then links l_1 and l_2 can be scheduled to simultaneously transmit without interfering with each other.

It must be noted that when the arrivals are given by a and the channel state is c, we have the following constraints on the set of feasible schedules:

$$\sum_{j=t}^{\tau_{lt}} s_{lj} \le a_{lt} \text{ for all } t \in \mathcal{T}_l^a,\, l \in \mathcal{L}, \tag{1}$$

$$s_{lt} = 0 \text{ for all } t \in \mathcal{T} \setminus \cup_{t\in\mathcal{T}_l^a}\{t,\ldots,\tau_{lt}\},\, l\in\mathcal{L}, \text{ and} \tag{2}$$

$$s_{lt} \le c_l \text{ for all } l \in \mathcal{L} \text{ and } t \in \mathcal{T}, \tag{3}$$

where (1) and (2) state that the number of successful inelastic transmissions is bounded by the number of available packet arrivals and that no packet should be scheduled after deadline expiration. Furthermore, (3) states that we cannot transmit more packets than what the channel state allows.

Denote the set of feasible schedules when the arrivals and channel state are a and c by $\mathcal{S}(a,c)$, capturing any interference constraints imposed by the network and satisfying (1), (2), and (3).

Our goal is to find $Pr(s|a,c)$ which is the probability of using schedule $s \in \mathcal{S}(a,c)$ when the arrivals are given by a and the channel state is c, subject to the constraint that the loss probability at link $l \in \mathcal{L}$ due to deadline expiry cannot exceed p_l.

Denoting by $\mu(a,c) = (\mu_l(a,c))_{l\in\mathcal{L}}$ the expected number of packets served when the arrivals and channel state are given by a and c, respectively, we have:

$$\mu_l(a,c) \le \sum_{s\in\mathcal{S}(a,c)} \sum_{t\in\mathcal{T}} s_{lt} Pr(s|a,c) \text{ for all } l \in \mathcal{L}.$$

Thus, the expected service at link $l \in \mathcal{L}$ is given by

$$\mu_l \overset{def}{=} E[\mu_l(a,c)].$$

Due to QoS constraints we need at all links

$$\mu_l \ge \lambda_l(1-p_l),$$

and to avoid trivialities, we assume that $\lambda_l(1-p_l) > 0$ for all $l \in \mathcal{L}$.

For notational simplicity, define the capacity region for fixed arrivals and channel state as

$$\mathcal{C}(a,c) \overset{def}{=} \left\{ \begin{array}{l} (\bar{\mu}_l)_{l\in\mathcal{L}} : \text{there exists } \bar{s} \in \mathcal{S}(a,c)_{\mathcal{CH}}, \\ \bar{\mu}_l \le \sum_{t\in\mathcal{T}} \bar{s}_{l,t} \end{array} \right\},$$

where $\mathcal{S}(a,c)_{\mathcal{CH}}$ is the convex hull of $\mathcal{S}(a,c)$.

Thus, the overall capacity region can be defined as follows:

$$\mathcal{C} \overset{def}{=} \left\{ \begin{array}{l} (\mu_l)_{l\in\mathcal{L}} : \text{there exists } (\bar{\mu}_l(a,c))_{l\in\mathcal{L}} \in \mathcal{C}(a,c) \\ \quad \text{for all } a,c \text{ and } \mu_l = E[\bar{\mu}_l(a,c)] \text{ for all } l \in \mathcal{L} \end{array} \right\}.$$

We will focus on the following static formulation for our problem, for some given vector $w \in \mathbb{R}_+^{|\mathcal{L}|}$:

$$\max_{\mu\in\mathcal{C}} \sum_{l\in\mathcal{L}} w_l \mu_l \tag{4}$$

subject to

$$\mu_l \ge \lambda_l(1-p_l) \text{ for all } l \in \mathcal{L}.$$

It must be noted that in the formulation we impose the condition that the service to all links have to meet their minimum QoS requirements. In order to allocate the additional bandwidth to flows beyond what is required to meet their needs, we can use the vector w to prioritize some links over others. Other uses for w have been explored in [17] for the case of scheduling both elastic and inelastic traffic. The related admission control problem, verifying a given traffic load is within the capacity region, is a very important and difficult problem. We leave this admission control problem for future research, and we will assume that the arrivals and loss probability requirements are feasible and thus the optimization problem has a solution μ^*.

IV. Dual Decomposition of the Static Problem

In this section we use duality theory to decompose the static optimization problem into simpler subproblems that will give us the ideas behind the dynamic algorithm.

Using the definition of the dual function [19], we have that

$$D(\delta) = \max_{\mu\in\mathcal{C}} \sum_{l\in\mathcal{L}} w_l\mu_l - \delta_l[\lambda_l(1-p_l) - \mu_l],$$

where

$$\delta^* \in \arg\min_{\delta_l \ge 0} D(\delta).$$

We are interested in finding μ^* but not the value $D(\delta^*)$, so the problem can be simplified as follows

$$\max_{\mu\in\mathcal{C}} \sum_{l\in\mathcal{L}} (w_l + \delta_l)\mu_l. \tag{5}$$

Since we are interested in solving the problem for non-negative values of δ_l, it must be the case that μ^* is as large as the constraints allow. Furthermore, since the objective function in (5) is linear, the problem can be decomposed into the following subproblems for fixed a and c:

$$\max_{s\in\mathcal{S}(a,c)} \sum_{l\in\mathcal{L}} (w_l + \delta_l) \sum_{t\in\mathcal{T}} s_{lt}.$$

The optimization problem can be solved using the following iterative algorithm, where k is the step index:

$$\tilde{s}^*(a,c,k) \in \arg\max_{s\in\mathcal{S}(a,c)} \sum_{l\in\mathcal{L}} [w_l + \delta_l(k)] \sum_{t\in\mathcal{T}} s_{lt}$$

$$\tilde{\mu}_l^*(k) = \sum_{a,c} \sum_{t\in\mathcal{T}} \tilde{s}_{lt}^*(a,c,k) Pr(c) Pr(a).$$

And the update equation for the Lagrange multipliers is given by

$$\delta_l(k+1) = \{\delta_l(k) + \epsilon[\lambda_l(1-p_l) - \tilde{\mu}_l^*(k)]\}^+,$$

where $\epsilon > 0$ is a fixed step-size parameter, and for any $\alpha \in \mathbb{R}$, $\alpha^+ \overset{def}{=} \max\{\alpha, 0\}$.

With the change of variables $\epsilon\hat{d}(k) = \delta(k)$, we rewrite the algorithm as

$$\tilde{s}^*(a, c, k) \in \operatorname*{arg\,max}_{s\in\mathcal{S}(a,c)} \sum_{l\in\mathcal{L}} [\frac{1}{\epsilon} w_l + \hat{d}_l(k)] \sum_{t\in\mathcal{T}} s_{lt} \qquad (6)$$

$$\tilde{\mu}_l^*(k) = \sum_{a,c} \sum_{t\in\mathcal{T}} \tilde{s}_{lt}^*(a, c, k) Pr(c) Pr(a),$$

with update equation:

$$\hat{d}_l(k + 1) = [\hat{d}_l(k) + \lambda_l(1 - p_l) - \tilde{\mu}_l^*(k)]]^+.$$

From the update equation we see that $\hat{d}_l(k)$ can be interpreted as a queue with $\lambda_l(1 - p_l)$ arrivals and $\tilde{\mu}_l^*(k)$ departures at step k.

V. Online Algorithm and Its Convergence Analysis

So far we have presented a dual decomposition for a static problem; however, the real network has stochastic arrivals and channel state. We will use the intuition from the decomposition in this section to develop an online algorithm that can cope with such changing conditions and prove its convergence properties.

A. Scheduler

We propose the following dynamic scheduling algorithm, where the arrivals and channel state in frame k are given by $a(k)$ and $c(k)$, respectively:

$$\tilde{s}^*(a(k), c(k), d(k)) \in \operatorname*{arg\,max}_{s\in\mathcal{S}(a(k),c(k))} \sum_{l\in\mathcal{L}} [\frac{1}{\epsilon} w_l + d_l(k)] \sum_{t\in\mathcal{T}} s_{lt}, \qquad (7)$$

with update equation

$$d_l(k + 1) = [d_l(k) + \tilde{a}_l(k) - I_l^*(a(k), c(k), d(k))]^+,$$

where

$$I_l^*(a(k), c(k), d(k)) = \sum_{t\in\mathcal{T}} \tilde{s}_{lt}^*(a(k), c(k), d(k))$$

and $\tilde{a}_l(k)$ is a binomial random variable with parameters $a_l(k)$ and $1 - p_l$. The quantity $\tilde{a}_l(k)$ can be generated by the network as follows: upon each packet arrival, toss a coin with probability of *heads* equal to $1 - p_l$, and if the outcome is *heads*, add a one to the deficit counter $d_l(k)$. This implementation for $\tilde{a}_l(k)$ was first suggested in [17].

Note that in (7) we make explicit the fact that the optimal scheduler is a function of $a(k)$, $c(k)$, and $d(k)$, for fixed w and ϵ. Also note that $d_l(k)$ can be interpreted as a virtual queue that keeps track of the deficit in service for link l to achieve a loss probability due to deadline expiry less than or equal to p_l. The idea of using a deficit counter was first used in [14] for the case of collocated networks with homogeneous delays, while the Lagrange multiplier interpretation allowed us to extend the result to general ad hoc networks and heterogeneous delays.

B. Convergence Results

We now summarize our results, showing that on average the online algorithm meets the QoS constraints, the total expected service deficit has a $O(1/\epsilon)$ bound, and the expected value of the objective is within $O(\epsilon)$ of the optimal value of the static problem (4). The proofs are omitted since they are similar to the ones presented in [17].

Theorem 1: If there exists a point $\mu(\Delta) \in \mathcal{C}/(1 + \Delta)$ for some $\Delta > 0$ such that

$$\mu_l(\Delta) \geq \lambda_l(1 - p_l) \text{ for all } l \in \mathcal{L},$$

then the total expected service deficit is upper-bounded by

$$\limsup_{k\to\infty} E\left[\sum_{l\in\mathcal{L}} d_l(k)\right] \leq B_1 + \frac{1}{\epsilon} B_2 \qquad (8)$$

for some positive constants B_1, B_2. Furthermore, the online algorithm fulfills all the QoS constraints. That is:

$$\liminf_{K\to\infty} E\left[\frac{1}{K}\sum_{k=1}^{K} I_l^*(a(k), c(k), d(k))\right] \geq \lambda_l(1 - p_l) \qquad (9)$$

for all $l \in \mathcal{L}$. ◇

It must be noted that (8) establishes that the deficit counters have a $O(1/\epsilon)$ bound, and follows directly from the fact that for a quadratic Lyapunov funcion the expected drift is negative but for a finite set of values for the deficit counters. Regarding (9), the theorem states that the arrival rate into the deficit counter is less than or equal to the service rate. The result is a consequence of the stability of the deficit counters. The idea behind the proof is that since there is an upper bound on the deficit counters, the difference between the cumulative service and the minimum required service is also bounded, so the long term time average fulfills the QoS requirements.

Note that $\sum_{k=1}^{K-1} \tilde{a}_l(k)$ denotes the cumulative minimum number of packets that need to be transmitted before deadline expiry at link l, before the beginning of frame K, to fulfill the QoS requirement. Thus, if we let $\theta_l(K)$ denote the cumulative deficit to fulfill the QoS constraint, we have the following result:

$$\theta_l(K) = \sum_{k=1}^{K-1} \tilde{a}_l(k) - I_l^*(a(k), c(k), d(k)) \leq d_l(K).$$

The reason for the inequality comes from the update equation of $d_l(K)$ and the fact that it is a non-negative variable. Therefore, one can use $d_l(K)$ as an upper-bound on the cumulative deficit in service to meet QoS constraints. When $d_l(K)$ is bounded, $d_l(K)/K \to 0$ as $K \to \infty$, so the QoS constraints are met.

We can also prove that our online algorithm is within $O(\epsilon)$ of the optimal value.

Theorem 2: For any $\epsilon > 0$ we have that

$$\limsup_{K\to\infty} E\left[\sum_{l\in\mathcal{L}} \{w_l\mu_l^* - \frac{w_l}{K}\sum_{k=1}^{K} I_l^*(a(k), c(k), d(k))\}\right] \leq B\epsilon$$

for some $B > 0$, where μ^* is a solution to (4), and $I^*(a(k), c(k), d(k))$ is obtained from the solution to (7). ◇

VI. Unknown Channel State, Per-Slot Feedback

Compared to the previous case, the per-slot feedback case is more complex due to the fact that we can use the feedback to update our decisions at every time slot. In this section we will first formulate the problem focusing on policies rather than on schedules, and we will show that no simple decomposition can be achieved for this case. However, we will prove that for a simple scenario a greedy solution can achieve the optimal solution.

A. Problem Formulation and Solution

In this section we will only highlight the differences between this case and the known channel state case.

As described in Section II, the channel at link $l \in \mathcal{L}$, time slot $t \in \mathcal{T}$, c_{lt}, is assumed to be a Bernoulli random variable with mean $\bar{c}_l$. Thus, instead of choosing a schedule for the entire frame, we will try to find a scheduling policy ρ that makes decisions at every time slot based on the feedback received. Note however that if the arrivals and channel state at a given frame are given by a and c respectively, policy ρ will generate a schedule by the end of the frame. We will denote by $s(\rho, a, c)$ such schedule.

We only focus on feasible policies, which are defined to be policies that generate a schedule that meets all interference constraints given by the network and fulfill the following constraints, for fixed ρ, a, and c:

$$s_{lt}(\rho, a, c) = 0 \text{ for all } t \in \mathcal{T}\backslash\cup_{t\in\mathcal{T}_l^a}\{t, \ldots, \tau_{lt}\},\ l \in \mathcal{L}, \quad (10)$$

$$s_{lt}(\rho, a, c) \leq 1 \text{ for all } l \in \mathcal{L} \text{ and } t \in \mathcal{T}, \text{ and} \quad (11)$$

$$\sum_{j=t}^{\tau_{lt}} c_{lj} s_{lj}(\rho, a, c) \leq a_{lt} \text{ for all } t \in \mathcal{T}_l^a,\ l \in \mathcal{L}. \quad (12)$$

Note that (10) specifies that a link should not be scheduled if there is no packet to be transmitted, (11) states that we cannot schedule more than a packet in a time slot since the channel is Bernoulli, and (12) specifies that a feasible policy cannot have more successful transmissions than the number of packets available.

We highlight the fact that since there is only a finite number of feasible schedules, then the set of feasible policies is finite. We will denote by $\mathcal{P}(a)$ the set of feasible policies that meet all interference constraints and that fulfill (10), (11), and (12), when arrivals are given by a.

Our goal is to find the probability distribution $Pr(\rho|a)$ of using policy $\rho \in \mathcal{P}(a)$ in a given frame when arrivals are given by a, such that the fraction of packets that miss the deadline at link l cannot exceed p_l. Thus, the expected service at link $l \in \mathcal{L}$ is subject to the following constraint

$$\mu_l \leq E\left[\sum_{\rho\in\mathcal{P}(a)}\sum_{t\in\mathcal{T}} c_{lt} s_{lt}(\rho, a, c) Pr(\rho|a)\right].$$

Therefore, the optimization problem is as follows, for a given vector $w \in \mathbb{R}_+^{|\mathcal{L}|}$:

$$\max_{\mu, Pr(\rho|a)} \sum_{l\in\mathcal{L}} w_l \mu_l \quad (13)$$

subject to

$$\mu_l \leq E\left[\sum_{\rho\in\mathcal{P}(a)}\sum_{t\in\mathcal{T}} c_{lt} s_{lt}(\rho, a, c) Pr(\rho|a)\right] \text{ for all } l \in \mathcal{L}$$

$$\mu_l \geq \lambda_l(1 - p_l) \text{ for all } l \in \mathcal{L}$$

$$Pr(\rho|a) \geq 0 \text{ for all } \rho \in \mathcal{P}(a), a$$

$$\sum_{\rho\in\mathcal{P}(a)} Pr(\rho|a) \leq 1 \text{ for all } a.$$

We will assume that the arrivals and loss probability requirements are feasible and thus the optimization problem has a solution μ^*.

Following the arguments in Section IV, we can develop the design ideas behind the following dynamic scheduler, assuming that at frame k the arrivals are given by $a(k)$ and the channel state by $c(k)$:

$$\tilde{\rho}^*(a(k), d(k)) \in \arg\max_{\rho\in\mathcal{P}(a(k))} \sum_{l\in\mathcal{L}}[\frac{1}{\epsilon}w_l + d_l(k)]\mu_l(\rho, a(k)) \quad (14)$$

with update equation

$$d_l(k+1) = [d_l(k) + \tilde{a}_l(k) - I_l^*(a(k), c(k), d(k))]]^+,$$

where

$$\mu_l(\rho, a(k)) = \sum_c \sum_{t\in\mathcal{T}} c_{lt} s_{lt}(\rho, a(k), c) Pr(c), \quad (15)$$

$$I_l^*(a(k), c(k), d(k)) = \sum_{t\in\mathcal{T}} c_{lt}(k) s_{lt}(\tilde{\rho}^*(k), a(k), c(k)),$$

$\tilde{\rho}^*(k) = \tilde{\rho}^*(a(k), d(k))$, and $\tilde{a}_l(k)$ is a binomial random variable with parameters $a_l(k)$ and $1 - p_l$.

We note that compared to the known channel case, the algorithm needs to know the probability distribution of c in order to make optimal decisions. From (14) we see that the duality approach does not give us a simple decomposition as in (6). The reason comes from the fact that even though per-slot feedback may help us to potentially increase the throughput, it also increases the complexity of the decision algorithm. However, it is important to highlight that (14) can be solved using dynamic programming [20]. To see that, define the expected utility when arrivals are given by a and there are T time slots remaining by

$$U(a, T) \overset{def}{=} \max_{\rho\in\mathcal{P}(a)} \sum_{l\in\mathcal{L}} \pi_l \mu_l(\rho, a),$$

where

$$\pi_l = \frac{1}{\epsilon}w_l + d_l.$$

Using (15), we can write the backwards equation of dynamic programming for the system:

$$U(a, T) = \max_{(s_{l1})}\left[\sum_{l\in\mathcal{L}} \pi_l \bar{c}_l s_{l1} + \sum_{\alpha\in\mathcal{A}} Pr(\alpha|a, s_{l1}) U(\alpha, T-1)\right]$$

where

$$\mathcal{A} \overset{def}{=} \{\alpha : 0 \leq \alpha_l \leq a_l \text{ for all } l \in \mathcal{L}\}$$

is the set of remaining number of packets at link l at the end of time slot 1, $Pr(\alpha|a, s_{l1})$ is the probability that the arrivals are given by α when there are $T-1$ time slots remaining, given that in the previous slot the arrivals were given by a and the chosen schedule in the first time slot was (s_{l1}), and by convention, let $U(\alpha, 0) = 0$.

Using the same proof techniques as in Section V-B, it can be proved that the scheduler meets all the QoS requirements, the total expected service deficits have an $O(1/\epsilon)$ bound, and the mean value of the objective is within $O(\epsilon)$ of the optimal value.

B. A Greedy Strategy for Collocated Networks

In this Section we will show that in a simple scenario a greedy algorithm can achieve the optimal solution with minimum complexity. To do that, we will focus our attention to collocated networks, where only one link is allowed to transmit at any given time slot. We will also assume that the channel state is independent between different time slots. Furthermore, we will assume that at every frame there is a single packet arrival at every link at the beginning of the frame, and that all the packets must be transmitted by the end of the frame. That is,

$$\mathcal{T}_l^a = \{1\},\ a_{l1} = 1, \text{ and } \tau_{l1} = T \text{ for all } l \in \mathcal{L}. \tag{16}$$

The key idea we will use in this section is that for a given frame when the deficit counters take value d, links will be prioritized in decreasing order of the priorities $[\frac{1}{\epsilon} w_l + d_l]\bar{c}_l$.

Definition 1: A *greedy policy* for collocated networks is a scheduling policy that at every time slot schedules a link with the highest priority $[\frac{1}{\epsilon} w_l + d_l]\bar{c}_l$ among the links that have a packet that remains to be transmitted. ◇

Theorem 3: The greedy scheduler is the optimal solution to (14) for collocated networks, when the arrivals are given by (16), and the channel state is independent between different time slots. ◇

For ease of readability, we defer the proof to the appendix. Due to the optimality of the policy, and following a similar development as in Section V-B, one can prove that the greedy algorithm meets all the QoS constraints, the total expected service deficits have an $O(1/\epsilon)$ bound, and the mean value of the objective is within $O(\epsilon)$ of the optimal value. We skip the proofs since they are analogous to the ones already presented.

The above theorem shows that the dual decomposition solution presented here recovers the solution for the special case of access-point networks presented in [14]. The contribution of this section is to show that the dual approach allows us to extend such results for very general ad hoc networks, arrivals, and for heterogeneous delays, and that [14] can be seen as a particular case of our general formulation.

VII. Simulations

The objective of the simulations is twofold: First, we will study the performance of a greedy algorithm that helps simplify the complexity of the proposed schedulers. Second, we study the performance of the schedulers for the different channel models we have studied in this paper.

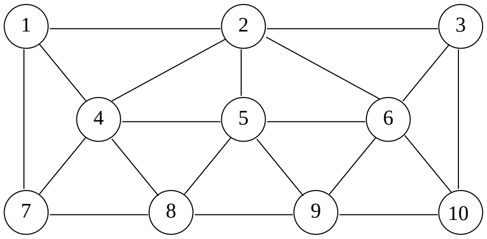

Fig. 1. Interference graph used in the simulations

We simulate a 10-link network such that its interference graph is given by Fig. 1, where each vertex of the graph represents a link and the edges represent the interference constraints. For example, if link 1 is scheduled, then links 2, 4, and 7 cannot be activated. This interference graph was first used in [17]. The constraint for the dropping probability due to deadline expiration is set to $p = 0.1$, the packet arrivals at every link are assumed to be Bernoulli random variables with mean λ packets/frame, and a frame has 3 time slots. Every channel is assumed to be a Bernoulli random variable with mean 0.96. We set the step-size parameter $\epsilon = 0.1$, while the link weights are set to $w_l = 6$ for all links. The simulation time is 10^5 frames.

As noted in (7) the scheduler needs to maximize the total sum weight for a frame, which leads to an exponential increase in the complexity of the algorithm with respect to the number of time slots in a frame. A natural question to ask is if it is necessary to optimize over a frame, or if a greedy strategy that independently maximizes the weighted summation over every time slot suffices, leading to an algorithm that increases its complexity linearly with the number of slots in a frame. In Figs. 2 and 3 we compare the optimal scheduler with such greedy algorithm. As it can be seen, the greedy algorithm incurs in a loss of throughput. The reason for this is that due to the interference constraints, greedy decisions in the first time slots can lead to suboptimal decisions later, which implies a smaller number of packets served. In practice, a greedy algorithm leads to much smaller complexity than the optimal algorithm. While it is sub-optimal, the throughput under the greedy algorithm is not substantially different compared with the optimal algorithm. As has been observed in the vast prior literature on maxweight algorithms (see, for example, [21]), greedy algorithms perhaps perform reasonably well because of the fact that they do take into account queue-length information, although they do not act on this information optimally. For the rest of the simulations we will then use the greedy algorithm as an approximation to study the performance of the proposed scheduling algorithms.

In Fig. 4 we compare the dropping probability for the different channel models we have studied. It must be noted that in the case of per-slot feedback we have the smallest dropping probability of both channel models. It must be noted that since the channel is assumed to vary from time slot to time slot, and since we have immediate feedback on the success of the transmission, once a packet is successfully transmitted another packet from a different unscheduled link can be transmitted in the remaining time slots. In the known channel case we

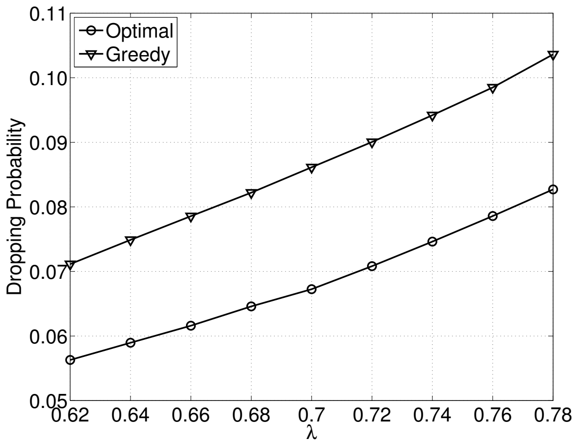

Fig. 2. Comparison of dropping probability for a greedy approximation

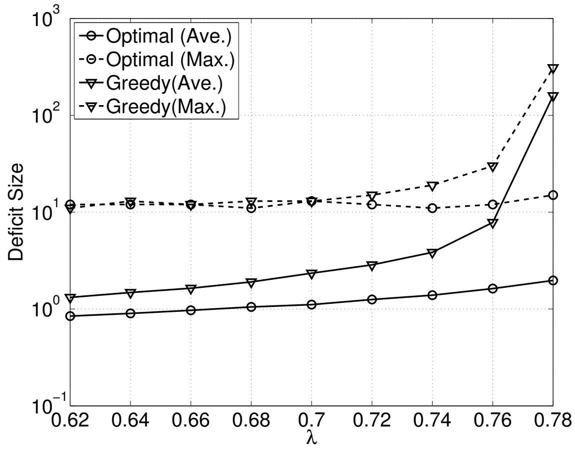

Fig. 3. Comparison of the average and maximum deficit size for a greedy approximation

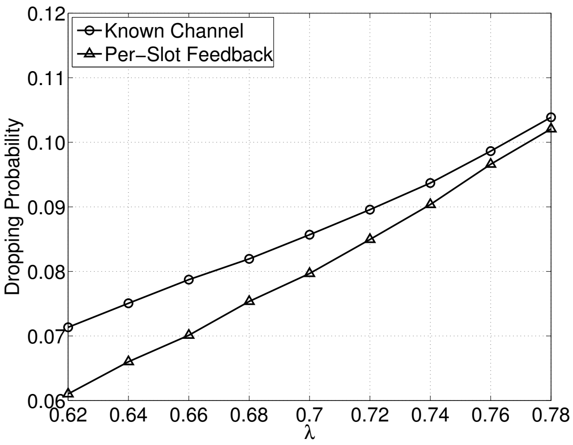

Fig. 4. Comparison of dropping probability for different channel models

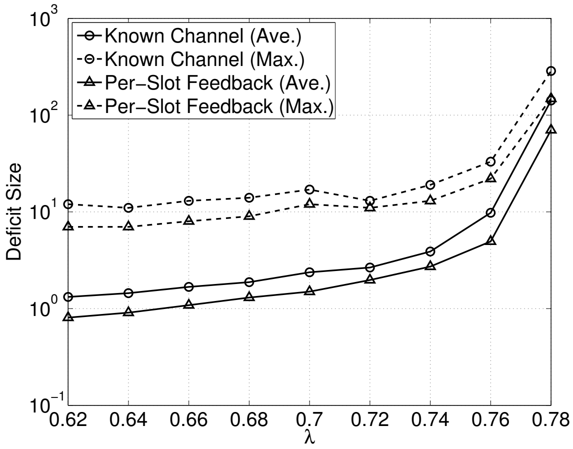

Fig. 5. Comparison of the average and maximum deficit size for different channel models

assume that the channel is fixed for a frame, and therefore if the channel is OFF for a link, that link will be not served during the entire frame.

Note in Fig. 5, where we compare the deficit size for the different channel models, that the deficit counters remain close to zero when the dropping probability is below 0.1, which means we are within the capacity region, and that the deficit only starts to build up when we are approaching the boundary of such region. To understand the intuition behind this result, remember that each deficit counter is associated with a Lagrange multiplier from the static problem formulation. When we are within the capacity region, the service allocated to a link is strictly greater than the minimum QoS requirement, leading to an inactive constraint that has associated a zero Lagrange multiplier. Only when we are close to the boundary of the capacity region, and the inequalities start to become active we will have non-zero Lagrange multipliers. As a consequence of this, the step-size parameter ϵ, which was introduced in the update equation for the Lagrange multipliers, only plays a role close to the boundary. We have corroborated this fact in our simulations, as can be seen in Figs. 6 and 7, where the mean arrival rate is $\lambda = 0.6$.

VIII. Conclusions

In this work we have presented an optimization formulation for the problem of scheduling real-time traffic in ad hoc networks under maximum delay constraints. The model allows for heterogeneous delay constraints and time-varying channels. Using duality theory and a decomposition approach, we presented an optimal scheduler and proved that it fairly allocates data rates to all links and guarantees that the delay requirements are met at every flow. We further studied the impact of feedback at every time slot on the complexity of the optimal algorithm, and showed that for a certain simple scenario a greedy strategy can achieve the optimal solution with low complexity, recovering the results of [14] for access-point networks.

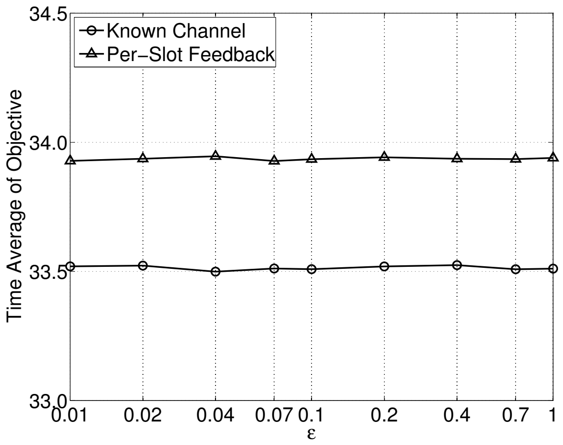

Fig. 6. Average objective for different values of ϵ

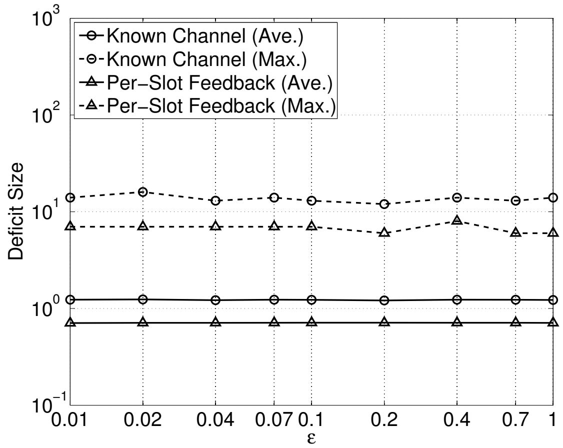

Fig. 7. Deficit size for the different values of ϵ

Appendix A
Proof of Theorem 3

The proof will use dynamic programming arguments: we will first note that if there is only one time slot remaining, the optimal decision is to schedule the link with the highest weight among the links that have a packet that remains to be transmitted, and then using induction we will prove that the best decision in any time slot is to schedule a backlogged link with the highest weight.

For simplicity in notation, define $\pi_l = \frac{1}{\epsilon}w_l + d_l$ for all $l \in \mathcal{L}$. Also define

$$U_\rho(\mathcal{L}, j) = \sum_{l\in\mathcal{L}} \sum_{c} \sum_{t=T-j+1}^{T} \pi_l c_{lt} s_{lt}(\rho, a, c) Pr(c)$$

to be the expected utility of policy ρ when there are j time slots remaining and the set of links that have a packet that needs to be transmitted is given by $\mathcal{L}$. Furthermore, define

$$U_\rho(\mathcal{L}, 0) = 0.$$

Finally, we will denote the greedy policy by g.

If there is only one time slot remaining, the optimal decision is to schedule one of the links that has the largest weight $\pi_l \bar{c}_l$ among the links that have a waiting packet, since this maximizes the expected utility. So the optimal decision is to use the greedy scheduler in the last time slot. Using induction, we assume that when there are j time slots remaining, it is optimal to use the greedy scheduler. We will prove that if there are $j+1$ time slots remaining then it is also optimal to use the greedy scheduler.

When there are $j+1$ time slots, we need to determine which link to schedule in the first slot, and then use the greedy scheduler for the remaining j time slots. Assume that the set of links that have a packet waiting to be transmitted is given by $\mathcal{L}$. If we schedule the link with the largest weight $\pi_l \bar{c}_l$, then the expected utility is given by

$$U_g(\mathcal{L}, j+1) = \pi_{l^*}\bar{c}_{l^*} + (1-\bar{c}_{l^*})U_g(\mathcal{L}, j) + \bar{c}_{l^*}U_g(\mathcal{L}\setminus\{l^*\}, j)$$

where

$$l^* \in \arg\max_{l\in\mathcal{L}} \pi_l \bar{c}_l.$$

If we decide to schedule link

$$\tilde{l} \notin \arg\max_{l\in\mathcal{L}} \pi_l \bar{c}_l,$$

then the expected utility is given by

$$\begin{aligned}
U_{\tilde{\rho}}(\mathcal{L}, j+1) =& \pi_{\tilde{l}}\bar{c}_{\tilde{l}} + (1-\bar{c}_{\tilde{l}})U_g(\mathcal{L}, j) + \bar{c}_{\tilde{l}}U_g(\mathcal{L}\setminus\{\tilde{l}\}, j)\\
=& \pi_{\tilde{l}}\bar{c}_{\tilde{l}} + \pi_{l^*}\bar{c}_{l^*} + (1-\bar{c}_{l^*})(1-\bar{c}_{\tilde{l}})U_g(\mathcal{L}, j-1)\\
&+ (1-\bar{c}_{l^*})\bar{c}_{\tilde{l}}U_g(\mathcal{L}\setminus\{\tilde{l}\}, j-1)\\
&+ \bar{c}_{l^*}(1-\bar{c}_{\tilde{l}})U_g(\mathcal{L}\setminus\{l^*\}, j-1)\\
&+ \bar{c}_{l^*}\bar{c}_{\tilde{l}}U_g(\mathcal{L}\setminus\{l^*, \tilde{l}\}, j-1)\\
=& \pi_{l^*}\bar{c}_{l^*} + (1-\bar{c}_{l^*})\Big[\pi_{\tilde{l}}\bar{c}_{\tilde{l}} + (1-\bar{c}_{\tilde{l}})U_g(\mathcal{L}, j-1)\\
&+ \bar{c}_{\tilde{l}}U_g(\mathcal{L}\setminus\{\tilde{l}\}, j-1)\Big]\\
&+ \bar{c}_{l^*}\Big[\pi_{\tilde{l}}\bar{c}_{\tilde{l}} + (1-\bar{c}_{\tilde{l}})U_g(\mathcal{L}\setminus\{l^*\}, j-1)\\
&+ \bar{c}_{\tilde{l}}U_g(\mathcal{L}\setminus\{l^*, \tilde{l}\}, j-1)\Big].
\end{aligned}$$

So in order to prove that $U_g(\mathcal{L}, j+1) \geq U_{\tilde{\rho}}(\mathcal{L}, j+1)$ it suffices to show that

$$\begin{aligned}
U_g(\mathcal{L}, j) \geq& \pi_{\tilde{l}}\bar{c}_{\tilde{l}} + (1-\bar{c}_{\tilde{l}})U_g(\mathcal{L}, j-1)\\
&+ \bar{c}_{\tilde{l}}U_g(\mathcal{L}\setminus\{\tilde{l}\}, j-1)
\end{aligned} \tag{17}$$

and

$$\begin{aligned}
U_g(\mathcal{L}\setminus\{l^*\}, j) \geq& \pi_{\tilde{l}}\bar{c}_{\tilde{l}} + (1-\bar{c}_{\tilde{l}})U_g(\mathcal{L}\setminus\{l^*\}, j-1)\\
&+ \bar{c}_{\tilde{l}}U_g(\mathcal{L}\setminus\{l^*, \tilde{l}\}, j-1).
\end{aligned} \tag{18}$$

From the assumption that the greedy scheduler is optimal when there are j slots remaining, it is clear that (17) and (18) are true, so the greedy scheduler is indeed optimal when there are $j+1$ slots remaining. ■

References

[1] A. Eryilmaz and R. Srikant, “Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control,” *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1333–1344, Dec. 2007.

[2] X. Lin and N. B. Shroff, “Joint rate control and scheduling in multihop wireless networks,” in *43rd IEEE Conference on Decision and Control (CDC)*, vol. 2, Atlantis, Paradise Island, Bahamas, Dec. 14–17, 2004, pp. 1484–1489.

[3] M. J. Neely, E. Modiano, and C.-P. Li, “Fairness and optimal stochastic control for heterogeneous networks,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 306–409, Apr. 2008.

[4] A. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,” *Queueing Systems*, vol. 50, no. 4, pp. 401–457, Aug. 2005.

[5] A. Eryilmaz and R. Srikant, “Joint congestion control, routing and mac for stability and fairness in wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[6] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, “Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks,” in *IEEE INFOCOM*, Barcelona, Catalunya, Spain, Apr. 23–29, 2006.

[7] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[8] X. Lin, N. B. Shroff, and R. Srikant, “A tutorial on cross-layer optimization in wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452– 1463, Jun. 2006.

[9] L. Georgiadis, M. J. Neely, and L. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” *Foundation and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[10] S. Shakkottai and R. Srikant, “Scheduling real-time traffic with deadlines over a wireless channel,” *Wireless Networks*, vol. 8, no. 1, pp. 13–26, Jan. 2002.

[11] V. Raghunathan, V. Borkar, M. Cao, and P. R. Kumar, “Index policies for real-time multicast scheduling for wireless broadcast systems,” in *IEEE INFOCOM*, Phoenix, AZ, USA, Apr. 13–18, 2008, pp. 1570–1578.

[12] A. Dua and N. Bambos, “Downlink wireless packet scheduling with deadlines,” *IEEE Trans. Mobile Comput.*, vol. 6, no. 12, pp. 1410–1425, Dec. 2007.

[13] Q. Liu, X. Wang, and G. B. Giannakis, “A cross-layer scheduling algorithm with QoS support in wireless networks,” *IEEE Trans. Veh. Technol.*, vol. 55, no. 3, pp. 839–847, May 2006.

[14] I.-H. Hou, V. Borkar, and P. R. Kumar, “A theory of QoS for wireless,” in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 19–25, 2009, pp. 486–494.

[15] I.-H. Hou and P. R. Kumar, “Admission control and scheduling for QoS guarantees for variable-bit-rate applications on wireless channels,” in *10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, New Orleans, LA, USA, May 18–21, 2009, pp. 175–184.

[16] ——, “Scheduling heterogeneous real-time traffic over fading wireless channels,” in *IEEE INFOCOM*, San Diego, CA, USA, Mar. 15–19, 2010.

[17] J. J. Jaramillo and R. Srikant, “Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic,” in *IEEE INFOCOM*, San Diego, CA, USA, Mar. 15–19, 2010.

[18] *Information technology – Generic coding of moving pictures and associated audio information: Systems*, ISO/IEC Std. 13 818-1, 2007.

[19] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Norwell, MA: Kluwer Academic Publishers, 2003.

[20] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Athena Scientific, 2005, vol. I.

[21] X. Lin and N. B. Shroff, “The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks,” in *IEEE INFOCOM*, Miami, FL, USA, Mar. 13–17, 2005.

Juan José Jaramillo (S ’06, M ’11) received his B.S. degree (summa cum laude) from Universidad Pontificia Bolivariana, Colombia, in 1998, his M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign in 2005 and 2010, respectively, all in electrical engineering. He is the recipient of a Fulbright fellowship. From 1999 to 2003 he worked at Empresas Publicas de Medellin in Colombia. He is currently with Iowa State University, where he is a Postdoctoral Research Associate at the Department of Electrical and Computer Engineering.

His research interests include communication networks and game theory.

R. Srikant (S ’90-M ’91-SM ’01-F ’06) received his B.Tech. from the Indian Institute of Technology, Madras in 1985, his M.S. and Ph.D. from the University of Illinois in 1988 and 1991, respectively, all in Electrical Engineering. He was a Member of Technical Staff at AT&T Bell Laboratories from 1991 to 1995. He is currently with the University of Illinois at Urbana-Champaign, where he is the Fredric G. and Elizabeth H. Nearing Endowed Professor in the Department of Electrical and Computer Engineering, and a Research Professor in the Coordinated Science Lab.

He was an associate editor of *Automatica* and the *IEEE Transactions on Automatic Control*, and is currently an associate editor of the *IEEE/ACM Transactions on Networking*. He has also served on the editorial boards of special issues of the *IEEE Journal on Selected Areas in Communications* and *IEEE Transactions on Information Theory*. He was the chair of the 2002 IEEE Computer Communications Workshop in Santa Fe, NM and was a program co-chair of IEEE INFOCOM, 2007. His research interests include communication networks, stochastic processes, queueing theory, information theory, and game theory.

Lei Ying (M’08) received his B.E. degree from Tsinghua University, Beijing, in 2001, his M.S. and Ph.D in Electrical Engineering from the University of Illinois at Urbana-Champaign in 2003 and 2007, respectively. During Fall 2007, he worked as a Postdoctoral fellow in the University of Texas at Austin. He is currently an Assistant Professor at the Department of Electrical and Computer Engineering at Iowa State University.

His research interest is broadly in the area of information networks, including wireless networks, mobile ad hoc networks, P2P networks, and social networks. He received a Young Investigator Award from the Defense Threat Reduction Agency (DTRA) in 2009, NSF CAREER Award in 2010, and is named Litton Assistant Professor at the Department of Electrical and Computer Engineering at Iowa State University for 2010-2011.