Structural Properties of LDP for Queue-Length Based Wireless Scheduling Algorithms

V. J. Venkataramanan and Xiaojun Lin

***Abstract*— In this paper, we are interested in wireless scheduling algorithms for the downlink of a single cell that can minimize the queue-overflow probability. Assuming that a sample-path large-deviation principle holds for the backlog process, we first study structural properties of the minimum-cost-path-to-overflow for a class of scheduling algorithms collectively referred to as the "α-algorithms." For a given $\alpha \geq 1$, the α-algorithm picks the user for service at each time that has the largest product of the transmission rate multiplied by the backlog raised to the power α. We show that when the overflow metric is appropriately modified, the minimum-cost-to-overflow under the α-algorithm can be achieved by a simple linear path, and it can be written as the solution of a vector-optimization problem. Using this structural property, we then show that when α approaches infinity, the α-algorithm asymptotically achieves the largest value of the minimum-cost-to-overflow under all scheduling algorithms.**

I. Introduction

Link scheduling is an important functionality in wireless networks due to both the shared nature of the wireless medium and the variations of the wireless channel over time. In the past, it has been demonstrated that, by carefully choosing the scheduling decision based on the channel state and/or the demand of the users, the system performance can be substantially improved (see the references in [1]). Most studies of scheduling algorithms have focused on optimizing the long-term average throughput of the users. Similarly, in the class of stability problems, the goal is to find scheduling algorithms that can stabilize the network at given offered loads, which also ensures that the long-term average service rate is no less than the arrival rate of each user. An important result along this direction is the development of the so-called "throughput-optimal" algorithms [2]. An algorithm is called *throughput-optimal* if, at any offered load that any other algorithm can stabilize the system, this algorithm can stabilize the system as well. Therefore, a throughput-optimal scheduling algorithm is optimal if we only impose stability constraints, i.e., it can stabilize the system over the largest set of offered loads.

While stability (and ensuring that the long-term service rate is no smaller than the arrival rate) is an important first-order metric of success, for many delay-sensitive applications it is far from sufficient. Note that a stability objective ensures that the packet delay does not increase to infinity. For real-time applications such as voice and video, we often need to ensure a stronger condition that the packet delay can be upper bounded with high probability. One approach to quantify the requirements of these delay-sensitive applications is to enforce constraints on the probability of queue overflow. In other words, we need to guarantee that the probability of each user's queue exceeding a given threshold is no greater than a target value.

In this paper, we are interested in scheduling algorithms that are optimal subject to the above type of delay constraints. We focus on the downlink of a single cell in a cellular network. The base-station serves multiple users. Due to interference, the base-station can only serve one user at a time. We assume that perfect channel information is available at the base-station. The ultimate question that we attempt to answer is the following: Is there a *delay-optimal* algorithm in the sense that, at any given offered load, the algorithm can achieve the smallest probability of queue-overflow. Note that if we impose a quality-of-service (QoS) constraint on each user in the form of an upper bound on the queue-overflow probability, then the above optimality condition will also imply that the algorithm can support the largest set of offered loads subject to the QoS constraint.

The above question has well-known to be a difficult one. The closest answer in the literature is provided in [3], where the author studied the problem in a large-deviation setting, and showed that the so-called "exponential-rule" is delay-optimal in the case with two-users. In a related result, it was shown that for the case when the service rate is fixed, the largest-weighted-delay-first (LWDF) algorithm is delay-optimal in a large-deviation setting [4], [5]. To the best of our knowledge, the general case for wireless networks with an arbitrary number of users is still open. Note that to study the queue-overflow probability, it is natural to use the large-deviation theory because the overflow probability of interest is often very small [6], [7]. The queue-overflow probability can then be mapped to the decay-rate of the tail-distribution of the queue, and the delay-optimal scheduling algorithm will correspond to the one that maximizes this delay-rate. Large-deviation theory has been successfully applied to wireline networks (see, e.g., [8]–[13]) and to wireless scheduling algorithms that only use the channel state to make the scheduling decisions [14]–[16]. However, when applied to wireless scheduling algorithms that use also the queue-length to make scheduling decisions, this approach encounters a significant amount of technical difficulty. Note than many scheduling algorithms of interest are of this latter flavor, i.e., they choose the user to serve based on both the channel state and the queue backlog. For example, the max-weight algorithm that is known to be throughput-optimal [17]

This work has been partially supported by the National Science Foundation through awards CCF-0635202, CNS-0643145, and CNS-0721484.

The authors are with School of ECE, Purdue University, West Lafayette, IN 47907. Email: {vvenkat,linx}@ecn.purdue.edu.

serves at each time the user with the largest product of the queue length and the service rate. Intuitively, this class of queue-length-based scheduling algorithms will have a lower queue-overflow probability compared to the queue-unaware algorithms because they make an effort to suppress longer queues. Indeed, the work in [18] has analytically shown the superiority of queue-length-based scheduling algorithms over queue-unaware algorithms for a symmetric case with ON-OFF channels. However, the technical difficulty associated with the queue-length-based scheduling algorithms is that the statistics of the service-rate process for each user is unknown (because now the service-rate process is tightly coupled with the backlog process, the channel variations, and the arrival process). In order to apply the large-deviation theory to queue-length-based scheduling algorithms, one has to use sample-path large-deviation, and formulate the problem as a multi-dimensional calculus-of-variations (CoV) problem for finding the "minimum-cost path to overflow." The decay-rate of the queue-overflow probability then corresponds to the cost of this path, which is referred to as the "minimum cost to overflow." Unfortunately, for many scheduling algorithms of interest, this multi-dimensional calculus-of-variations problem is very difficult to solve. In the literature, only some restricted cases have been solved: Either restricted problem structures are assumed (e.g., symmetric users and ON-OFF channels [18]), or the size of the system is very small (only two users) [19]. Due to the above difficulty, the question of finding the optimal wireless scheduling algorithms under delay constraints becomes very challenging.

In this paper, we provide a number of results along this direction. Assuming that a sample-path large-deviation principle holds, we study the structural property of the minimum-cost-path-to-overflow for a class of queue-length-based scheduling algorithms. In particular, we show that when the form of the overflow threshold is appropriately modified, at least one of the minimum-cost-path-to-overflow is linear. This result allows us to convert the calculus-of-variations problem (of sample-path large-deviation) to a vector-optimization problem. Using this structure property, we then show the main result of the paper that, as one of the parameters approaches infinity, these class of queue-length-based scheduling algorithms will asymptotically achieve the largest minimum-cost-to-overflow among all scheduling algorithms. As an immediate corollary of this result, we can show that with the ON-OFF channel model, the max-weight scheduling algorithm is optimal.

The rest of paper is organized as follows. We first present the system model and the class of queue-length-based scheduling algorithms (referred to as α-algorithms) in Section II. In Section III, we provide an upper bound on the minimum-cost-to-overflow for any scheduling algorithm. We then study the structural properties of the minimum-cost-path-to-overflow for α-algorithms in Section IV. Then in Section V, we prove the main result that, as the parameter α approaches infinity, this class of scheduling algorithms asymptotically achieve the largest possible value of the minimum-cost-to-overflow. Then we conclude.

II. The System Model and Assumptions

We consider the downlink of a single cell in which a base-station serves N users. We assume a slotted system, and we assume that the state of the channel at each time slot is *i.i.d* from one of M possible states. Let $C(t)$ denote the state of the channel at time $t = 1, 2, \ldots,$ and let $p_j = \mathbf{P}[C(t) = j], \quad j = 1, 2, \ldots, M$. Let $\vec{p} = [p_1, ..., p_M]$. We assume that the base-station can serve one user at a time. Let F_m^i denote the service rate for user i when it is picked for service at state m.

We assume that data for user i arrives as fluid at a constant rate λ_i. Let $\vec{\lambda} = [\lambda_1, \ldots, \lambda_N]$. Let $X_i(t)$ denote the backlog of user i at time t, and let $\vec{X}(t) = [X_1(t), \ldots, X_N(t)]$. In general, the decision of picking which user to serve is a function of the global backlog $\vec{X}(t)$ and the channel state $C(t)$. Let $U(t)$ denote the index of the user picked for service at time t. The evolution of the backlog for each user i is then given by

$$X_i(t+1) = [X_i(t) + \lambda_i - \sum_{m=1}^{M} \mathbf{1}_{\{C(t)=m, U(t)=i\}} F_m^i]^+ \quad (1)$$

where $[\cdot]^+$ denotes the projection to $[0, +\infty)$. Note that $\sum_{m=1}^{M} \sum_{i=1}^{N} \mathbf{1}_{\{C(t)=m, U(t)=i\}} = 1$ since only one user can be served at a time.

One particular class of scheduling algorithms that we will study are collectively referred to as the "α-algorithms", where α is a parameter that takes values from the set of natural numbers. Given α, the behaviour of the algorithm is as follows. When the backlog of the users is $\vec{X}(t)$ and the state of the channel is $C(t) = m$, the algorithm chooses to serve the user i for which the product $X_i^{\alpha}(t) F_m^i$ is the largest. If there are several users that achieve the largest $X_i^{\alpha}(t) F_m^i$ together, one of them is chosen arbitrarily. It is well-known that this class of algorithms are throughput-optimal, i.e. they can stabilize the system at the largest set of offer-loads $\vec{\lambda}$ [17].

Consider the system when it is operated at a given offered load and is stable under a given scheduling algorithm. In this paper, we are interested in the probability that the maximum backlog among the users exceeds a certain threshold B, i.e.,

$$\mathbf{P}[\max_i X_i(0) \geq B]. \quad (2)$$

Note that the probability in (2) is equivalent to a delay-violation probability when the arrival rates λ_i are constant, because the two types of events are related by (see [18], [20])

$$\mathbf{P}[\text{Delay at link } i \geq d_i] = \mathbf{P}[X_i(0) \geq \lambda_i d_i].$$

In this paper, we will be interested in scheduling algorithms that minimize (2), at a given offered $\vec{\lambda}$.

The problem of calculating the exact probability $\mathbf{P}[\max_i X_i(0) \geq B]$ is often mathematically intractable. In this paper, we are interested in using large-deviation techniques to compute estimates of this probability. Specifically, we are interested in those cases when the following limit exists.

$$\lim_{B\to\infty}\frac{1}{B}\log\mathbf{P}[\max_i X_i(0)\geq B] = -I_0(\vec{\lambda}). \tag{3}$$

(We will discuss how to compute $I_0(\vec{\lambda})$ using sample-path large-deviation and the corresponding assumptions in Section II-A). Note that if Equation (3) holds, it implies that, when B is large, the overflow probability can be approximated as

$$\mathbf{P}[\max_i X_i(0)\geq B] \approx \exp(-BI_0(\vec{\lambda})).$$

Thus, the scheduling algorithm that minimizes the overflow probability corresponds to the one that maximizes the decay-rate $I_0(\vec{\lambda})$.

A. Sample-Path Large Deviation

We next describe the sample-path large-deviation setting used to compute $I_0(\vec{\lambda})$. We follow the convention in [18], [21]. Use $B>0$ also as a scaling factor. For a large enough T, define the scaled empirical measure process on the time interval $[-T,0]$ as

$$s_j^B(t) = \frac{1}{B}\sum_{l=0}^{B(T+t)} \mathbf{1}_{\{C(l)=j\}},$$

for $t=\frac{k}{B}-T$, $k=0,...,BT$, and by linear interpolation otherwise. Note that, in the above definition, we have scaled both the time and the magnitude. The quantity $s_j^B(t)$ can be interpreted as the sum of the (scaled) time in $[-T,t]$ that the system is at state j. Further, it is easy to check that $\sum_{j=1}^M s_j^B(t) = t+T$ for all $t\in[-T,0]$. Let $\vec{s}^B(t) = [s_1^B(t),...s_M^B(t)]$. Analogously, define the scaled backlog process as,

$$x_i^B(t) = \frac{1}{B}X_i(B(T+t))$$

for $t=\frac{k}{B}-T$, $k=0,...,BT$, and by linear interpolation otherwise. Let $\vec{x}^B(t) = [x_1^B(t),...,x_N^B(t)]$. Note that the backlog process $\vec{x}^B(t)$ is related to the empirical measure process $s_j^B(t)$ by

$$\begin{aligned} & x_i^B(t+\frac{1}{B}) \\ = & \left[x_i^B(t)+\frac{\lambda_i}{B}\right. \\ & \left. -\sum_{m=1}^M (s_j^B(t)-s_j^B(t-\frac{1}{B}))\mathbf{1}_{\{U(B(T+t))=i\}}F_m^i\right]^+ . \end{aligned} \tag{4}$$

Thus, given a particular initial condition $\vec{x}^B(-T)$, Equation (4) defines a mapping $\mathbf{f}^B$ from the empirical measure process $\vec{s}^B(t)$ to the backlog process $\vec{x}^B(t)$. Further, although we have assumed $\vec{s}^B(t)$ to be piecewise linear to begin with, the definition of the mapping $\mathbf{f}^B$ can be naturally extended to all absolute continuous functions $\vec{s}^B(t)$.

For any $\vec{\phi}\geq 0$ and $\sum_{j=1}^M \phi_j = 1$, define $H(\vec{\phi}|\vec{p}) = \sum_{j=1}^M \phi_j \log\frac{\phi_j}{p_j}$. The sequence of empirical measure processes $\vec{s}^B(t)$ is known to satisfy a sample-path large deviation principle [7, p176] with large-deviation rate-function $I_s^T(\vec{s}(\cdot))$ given as follows:

$$I_s^T(\vec{s}(\cdot)) = \int_{-T}^0 H(\vec{\phi}(t)|\vec{p})dt,$$

if $\vec{s}(t)$ is absolute continuous and component-wise non-decreasing on $[-T,0]$, $\vec{s}(-T)=0$, and $\sum_{j=1}^M s_j(t)=t+T$ for all t; where $\vec{\phi}(t) = \frac{d}{dt}\vec{s}(t)$ (Note that $\vec{\phi}(t)$ is well defined almost everywhere on $[-T,0]$ since $\vec{s}(t)$ is absolute continuous on $[-T,0]$). Otherwise,

$$I_s^T(\vec{s}(\cdot)) = +\infty.$$

Such a large-deviation principle means that, for any set Γ of trajectories on $[-T,0]$ that is a *continuity* set [7, p5] according to the *essential supremum norm* [7, p176, p352], the probability that the sequence of empirical measure processes $\vec{s}^B(t)$ falls into Γ must satisfy

$$\lim_{B\to\infty}\frac{1}{B}\log\mathbf{P}[\vec{s}^B(\cdot)\in\Gamma] = -\inf_{\vec{s}(\cdot)\in\Gamma} I_s^T(\vec{s}(\cdot)). \tag{5}$$

In this paper, we assume that a sample-path large-deviation principle also holds for the sequence of backlog processes $\vec{x}^B(t)$. Specifically, we adopt the following assumptions:

A) As $B\to\infty$, the sequence of mappings $\mathbf{f}^B$ has a limiting mapping f that also maps any absolute continuous empirical measure process $\vec{s}(t)$ to a backlog processes $\vec{x}(t)$.

B) The mapping f is unique and is continuous with respect to appropriately-chosen topologies of the space of empirical measure processes and the space of the backlog processes.

C) The sequence of mappings $\mathbf{f}^B$ are *exponentially equivalent* to f [7, p130].

If these assumptions hold, then for any sequence of backlog processes that start from $\vec{x}^B(-T)=0$, we can invoke the contraction principle [7, p131] and obtain a sample-path large-deviation principle for the sequence of backlog processes $\vec{x}^B(t)$ with large-deviation rate-function given by:

$$I_x^T(\vec{x}(\cdot)) \quad = \inf_{\{\vec{s}(\cdot):\vec{x}(\cdot)=f(\vec{s}(\cdot))\}}\left\{\int_{-T}^0 H(\vec{\phi}(t)|\vec{p})dt\right\}$$

where $\vec{\phi}(t) = \frac{d}{dt}\vec{s}(t)$, and the infimum is taken over all empirical measure processes $\vec{s}(\cdot)$ that map (under the mapping f) to the same backlog process $\vec{x}(\cdot)$ given that $\vec{x}(-T)=0$. (We refer the readers to [21] for cases when these assumptions hold.)

Define an overflow metric as a function $h(\vec{x})$ such that $h(\vec{0})=0$, $h(B\vec{x})=Bh(\vec{x})$, and $h(\vec{x})$ is component-wise increasing. An overflow metric of the form $h(\vec{x})=\max_i x_i$, will be consistent with the queue-overflow threshold defined earlier. However, later we will also use other overflow metrics. The event of queue overflow is then represented by $h(\vec{x}^B(0)) \geq 1$. As $B \to \infty$, we have,

$$\begin{aligned} & -\lim_{B\to\infty} \frac{1}{B} \log \mathbf{P}[h(\vec{x}^B(0)) \geq 1] \\ = \quad & \inf\{I_x^T(\vec{x}(\cdot))| \text{ over all trajectories } \vec{x}(\cdot) \text{ that} \\ & \text{go from } \vec{x}(-T) = 0 \text{ for some } T > 0 \\ & \text{to } h(\vec{x}(0)) = 1\}. \qquad (6) \end{aligned}$$

The trajectory that attains the infimum in (6) is often called the *most likely path to overflow*. The value of the infimum itself is often called the *minimum cost to overflow*. Note that $I_0(\vec{\lambda})$ in (3) corresponds to (6) when $h(\vec{x}) = \max_i x_i$.

In the rest of the paper, our goal is to find scheduling algorithms that can achieve the largest value of $I_0(\vec{\lambda})$ (i.e., the largest value of the minimum-cost-to-overflow) at a given offered load $\vec{\lambda}$.

III. An upper bound on $I_0(\vec{\lambda})$

We first provide an upper bound on $I_0(\vec{\lambda})$ over all scheduling algorithms. Then, in Section V, we will show that the α-algorithm asymptotically achieves this upper bound as $\alpha \to \infty$, and hence is asymptotically optimal.

A. Definitions

Given a scheduling algorithm A (e.g., an "α-algorithm"), and an overflow metric $h(\cdot)$, let Ψ_A be the set of all possible trajectories under scheduling algorithm A. Precisely, each element of Ψ_A is a triplet $(\vec{\phi}(\cdot), \vec{x}(\cdot), T)$ such that $T > 0$, $\vec{\phi}(t) = \frac{d}{dt}\vec{s}(t)$ where $\vec{s}(\cdot)$ is an instance of the empirical measure process, $\vec{x}(-T) = 0$, and $x(t), \quad t \in [-T, 0]$, is the corresponding backlog process governed by the scheduling algorithm A . For ease of exposition, we use $\mathcal{F}(\Psi_A, h)$ to denote the calculus-of-variations problem in (6), i.e.,

$$\begin{aligned} \mathcal{F}(\Psi_A, h) \triangleq \inf_{\vec{\phi}(t),T} \quad & \int_{-T}^{0} H(\vec{\phi}(t)|\vec{p})dt \\ \text{subject to} \quad & (\vec{\phi}(\cdot), \vec{x}(\cdot), T) \in \Psi_A \qquad (7) \\ & h(\vec{x}(0)) = 1 \qquad (8) \\ & \vec{x}(-T) = 0. \qquad (9) \end{aligned}$$

In particular, we use $\mathcal{F}(\Psi_A, \max)$ to denote the case when $h(\vec{x}) = \max_i x_i$. Let $\Psi_A^* \subseteq \Psi_A$ be defined as follows

$$\Psi_A^* = \left\{(\vec{\phi}(\cdot), \vec{x}(\cdot), T) \in \Psi_A \text{ such that } \frac{d}{dt}\vec{\phi}(t) = \vec{0}\right\},$$

i.e., it contains all trajectories that correspond to a linear empirical measure process $\vec{s}(t)$. We can similarly define $\mathcal{F}(\Psi_A^*, h)$ where the constraints set Ψ_A in (7) is replaced by Ψ_A^*.

We define

$$\begin{aligned} \acute{w}(\vec{\phi}) \triangleq \min_{\vec{\phi},\vec{x}} \quad & \max_i(x_i) \\ \text{subject to} \quad & x_i = \left[\lambda_i - \sum_{m=1}^{M} \mu_m^i F_m^i\right]^+ \text{ for all } i \\ & \mu_m^i \geq 0, \sum_{i=1}^{N} \mu_m^i = \phi_m \text{ for all } m, \quad (10) \end{aligned}$$

and let

$$I_{opt} \triangleq \inf_{\vec{\phi}} \frac{H(\vec{\phi}|\vec{p})}{\acute{w}(\vec{\phi})}.$$

The following theorem states that I_{opt} is an upper bound on $I_0(\vec{\lambda})$ for any scheduling algorithm[*].

Theorem 1: For any scheduling algorithm A, $\mathcal{F}(\Psi_A, \max) \leq I_{opt}$.

Proof: First, note that by definition, $\Psi_A^* \subseteq \Psi_A$. This fact leads to the conclusion that $\mathcal{F}(\Psi_A, \max) \leq \mathcal{F}(\Psi_A^*, \max)$ since the constraint set in the optimization problem on the right hand side is smaller. Thus, it suffices to show the following

$$\mathcal{F}(\Psi_A^*, \max) \leq I_{opt} = \inf_{\vec{\phi}} \frac{H(\vec{\phi}|\vec{p})}{\acute{w}(\vec{\phi})}.$$

Consider any trajectory $(\vec{\phi}(t), \vec{x}(t), T)$ in the feasible region of $\mathcal{F}(\Psi_A^*, \max)$. Recall that $\vec{\phi}(t)$ is a constant by definition of Ψ_A^*. Denote $\vec{\phi}(t) = \vec{\phi}$. By (9), $\vec{x}(-T) = 0$. Further, by the queue-evolution equation (4) we have the following inequality, $x_i(0) \geq T[\lambda_i - \sum_{m=1}^{M} \mu_m^i F_m^i]^+$, where by μ_m^i we denote the average fraction of time in $[-T, 0]$ that the user i is served and the channel state is m. Finally, by (8) we know that $\max_i x_i(0) = 1$. We thus have

$$1 = \max_i x_i(0) \geq T \max_i([\lambda_i - \sum_{m=1}^{M} \mu_m^i F_m^i]^+) \geq T\acute{w}(\vec{\phi})$$

$$\Rightarrow TH(\vec{\phi}|\vec{p}) \leq \frac{H(\vec{\phi}|\vec{p})}{\acute{w}(\vec{\phi})}.$$

Note that $TH(\vec{\phi}|\vec{p})$ is precisely the cost of the trajectory. Since this inequality holds for all trajectories in Ψ_A^*, we have

$$\mathcal{F}(\Psi_A^*, \max) \leq I_{opt}.$$

■

IV. Structural Properties of the Minimum-Cost-Path-to-Overflow for α -algorithms

We next turn our attention to the α-algorithms. Our ultimate goal is to show in Section V that the α-algorithms asymptotically achieve the minimum-cost-to-overflow equal to I_{opt}. In this section, we first derive some structural

[*] This upper bound is equivalent to the one in [3].

properties of the minimum-cost-path-to-overflow under α-algorithms. Note that the calculus-of-variations problem in (6) and (9) with the overflow metric $h(\vec{x}) = \max_i x_i$ is often very difficult to solve. In general, the minimum-cost-path-to-overflow may not be of a simple linear form. The trick that we use here is to modify the overflow metric to one that is tailored to the scheduling algorithm. In particular, for the α-algorithm, we use the overflow metric

$$h(\vec{x}) = V_\alpha(\vec{x}) \triangleq \left(\sum_{i=1}^{N} x_i^{\alpha+1}\right)^{\frac{1}{\alpha+1}}.$$

Note that $V_\alpha(\vec{x})$ is well-known to be the Lyapunov function for proving that the α-algorithm is throughput-optimal. Thus we will refer to $V_\alpha(\vec{x})$ as the Lyapunov overflow metric, and refer to $h(\vec{x}) = \max_i x_i$ as the max-queue overflow metric. The connection between $V_\alpha(\cdot)$ and $\max_i x_i$ will be clear in Section V.

With the overflow metric $V_\alpha(\vec{x})$, the calculus-of-variations problem for finding the minimum-cost-to-overflow is represented by $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha)$.

A. A Lower bound on the minimum-cost-to-overflow

We first provide a lower bound on $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha)$. We start with a property of the limiting mapping f that maps the empirical measure process $\vec{s}(t)$ to the backlog process $\vec{x}(t)$. Note that according to the definition of $\vec{x}(t)$ and $\vec{s}(t)$, they are both Lipschitz-continuous, and hence are differentiable almost everywhere. For any time t such that both $\vec{x}(t)$ and $\vec{s}(t)$ are differentiable, the following properties can be shown: There must exist $\mu_m^i(t) \geq 0$ such that $\sum_{i=1}^{N} \mu_m^i(t) = \phi_m(t)$ and $\dot{x}_i(t) = [\lambda_i - \sum_{i=1}^{N} \mu_m^i(t) F_m^i]_{x_i(t)}^{+}$, where we have used the notation

$$[u]_v^+ = \begin{cases} u & \text{if } v > 0 \text{ or } u \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Recall that $\phi_m(t) = \frac{d}{dt} s_m(t)$ can be viewed as the fraction of time the system is in state m in an interval $[Bt, B(t+\delta t)]$ immediately after t. $\mu_m^i(t)$ can then be viewed as the fraction of time that user i is served and the system is in state m within such an interval. In addition, the following lemma can be shown.

Lemma 2:

$$\mu_m^i(t) = 0 \text{ if } x_i^\alpha(t) F_m^i < \max_k x_k^\alpha(t) F_m^k.$$

Proof: This can be shown by noting that if $x_i^\alpha(t) F_m^i < \max_k x_k^\alpha(t) F_m^k$, then for all sufficiently large B, $(x_i^B(s))^\alpha F_m^i < \max_k (x_k^B(s))^\alpha F_m^k$ holds for an interval $s \in [Bt, B(t+\delta t)]$ immediately after t. Hence, user i will not be picked for transmission over this entire interval. We can thus show that $\mu_m^i(t) = 0$. ■

We now use the Lyapunov function approach in [22] to derive a lower bound on $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha)$. First, define a local rate function of $\vec{x}(t)$ as:

$$\begin{aligned} l(\vec{x}, \vec{y}) = \inf_{\vec{\phi}} \quad & H(\vec{\phi}|\vec{p}) \\ \text{subject to} \quad & y_i = [\lambda_i - \sum_{m=1}^{M} \mu_m^i F_m^i]_{x_i}^{+} \text{ for all } i \\ & \mu_m^i \geq 0 \text{ and } \sum_{i=1}^{N} \mu_m^i = \phi_m \text{ for all } m \\ & \mu_m^i = 0 \text{ if } x_i^\alpha F_m^i < \max_k x_k^\alpha F_m^k. \end{aligned}$$

Note that $l(\vec{x}, \vec{y})$ denotes how likely the trajectory $\vec{x}(\cdot)$ can move in the direction $\frac{d}{dt}\vec{x}(t) = \vec{y}$ immediately after t, given $\vec{x}(t) = \vec{x}$. Using Lemma 2 we thus have

$$\begin{aligned} \mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha) \geq \inf_{T} \quad & \int_{-T}^{0} l(\vec{x}(t), \dot{\vec{x}}(t)) dt \\ \text{subject to} \quad & (\vec{\phi}(\cdot), \vec{x}(\cdot), T) \in \Psi_A \\ & V_\alpha(\vec{x}(0)) = 1 \\ & \vec{x}(-T) = 0. \end{aligned}$$

Further, letting $V(t) = V_\alpha(\vec{x}(t))$, we can define the local rate-function of $V(t)$ as

$$\begin{aligned} l_V(v, w) = \inf_{\vec{x}, \vec{y}} \quad & l(\vec{x}, \vec{y}) \\ \text{subject to} \quad & V_\alpha(\vec{x}) = v \\ & \left[\frac{\partial}{\partial \vec{x}} V_\alpha(\vec{x})\right]^T . \vec{y} = w. \end{aligned}$$

Then,

$$\begin{aligned} \mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha) \geq \inf_{T} \quad & \int_{-T}^{0} l_v(V(t), \dot{V}(t)) dt \\ \text{subject to} \quad & V(-T) = 0, V(0) = 1. \end{aligned} \quad (11)$$

Note that the right-hand-side is a one-dimension calculus-of-variations problem that is much easier to solve. For α-algorithms, if y_i and μ_m^i satisfy the constraints of $l(\vec{x}, \vec{y})$, then

$$\begin{aligned} & \left[\frac{\partial}{\partial \vec{x}} V_\alpha(\vec{x})\right]^T . \vec{y} \\ = \quad & \left(\sum_{i=1}^{N} x_i^{\alpha+1}\right)^{-\frac{\alpha}{\alpha+1}} \left[\sum_{i=1}^{N} x_i^\alpha (\lambda_i - \sum_{m=1}^{M} \mu_m^i F_m^i)\right] \\ = \quad & \left(\sum_{i=1}^{N} x_i^{\alpha+1}\right)^{-\frac{\alpha}{\alpha+1}} \left[\sum_{i=1}^{N} x_i^\alpha \lambda_i \right. \\ & \left. - \sum_{m=1}^{M} \phi_m \max_i x_i^\alpha F_m^i \right]. \end{aligned} \quad (12)$$

Hence, the local rate-function of $V(t)$ can be rewritten as

$$l_V(v, w) = \inf_{\vec{\phi}, \vec{x}} \quad H(\vec{\phi}|\vec{p})$$

$$\begin{aligned}\text{subject to} \quad & \left(\sum_{i=1}^N x_i^{\alpha+1}\right)^{-\frac{\alpha}{\alpha+1}} \left[\sum_{i=1}^N x_i^{\alpha} \lambda_i \right.\\ & \left. - \sum_{m=1}^M \phi_m \max_i x_i^{\alpha} F_m^i\right] = w\\ & \left(\sum_{i=1}^N x_i^{\alpha+1}\right)^{\frac{1}{\alpha+1}} = v.\end{aligned}$$

It is easy to show that $l_V(v,w)$ is independent of the value of v, i.e., $l_V(v,w) = l_V(1,w)$ for all $v \neq 0$. Let $l(w) \triangleq l_V(1,w)$. Then the solution to the calculus-of-variations problem on the right-hand-side of (11) is given by [6, p520]

$$\begin{aligned} J_1 \quad \triangleq \quad & \min_{w \geq 0} \frac{1}{w} l(w)\\ = \min_{\vec{\phi}, \vec{x}, w \geq 0} \quad & \frac{1}{w} H(\vec{\phi} || \vec{p})\\ \text{subject to} \quad & \left[\sum_{i=1}^N x_i^{\alpha} \lambda_i - \sum_{m=1}^M \phi_m \max_i x_i^{\alpha} F_m^i\right] = w\\ & \left(\sum_{i=1}^N x_i^{\alpha+1}\right)^{\frac{1}{\alpha+1}} = 1. \end{aligned} \tag{13}$$

We thus obtain the following result.

Lemma 3: The minimum cost to overflow $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha)$ must be no smaller than J_1.

B. Attainability of the Lower-bound J_1

In this subsection, we show that the lower bound J_1 is attainable with a simple linear trajectory $\vec{s}(t) = (t+T)\vec{\phi}, \quad t \geq -T$. Note that the solution of (13) will produce a $\vec{\phi}^*$ (it is easy to verify that such a $\vec{\phi}^*$ always exists). If this $\vec{\phi}^*$ can in fact map to a trajectory that starts from $\vec{x}(-T) = 0$ and overflows at $t = 0$, then the minimum-cost-to-overflow $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha)$ must be no larger than the cost of this trajectory $J_2 = TH(\vec{\phi}^* | \vec{p})$. Further, if $J_2 = J_1$, then we can conclude that $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha) = J_1$. We next show that this is indeed the case.

Towards this end, we first show that for each linear empirical measure process $\vec{s}(t) = (t+T)\vec{\phi}, \quad t \geq -T$, there exists a unique trajectory $\vec{x}(t)$ starting from $\vec{x}(-T) = 0$. We will need the following lemma.

Lemma 4: (a) Given $\vec{\phi}$, the optimal values of the following two problems are the same.

$$\begin{aligned} a(\vec{\phi}) = \max_{\vec{x} \geq 0} \quad & \left[\sum_{i=1}^N x_i^{\alpha} \lambda_i - \sum_{m=1}^M \phi_m \max_i x_i^{\alpha} F_m^i\right]\\ \text{subject to} \quad & \sum_{i=1}^N x_i^{\alpha+1} \leq 1, \end{aligned}$$

and

$$\begin{aligned} b(\vec{\phi}) = \min_{\vec{y} \geq 0} \quad & \left(\sum_{i=1}^N y_i^{\alpha+1}\right)^{\frac{1}{\alpha+1}}\\ \text{subject to} \quad & y_i = [\lambda_i - \sum_{m=1}^M F_m^i \mu_m^i]^+ \text{ for all } i\\ & \mu_m^i \geq 0, \sum_{i=1}^N \mu_m^i = \phi_m \text{ for all } m. \end{aligned}$$

(b) The optimizer $\vec{x}^*$ for $a(\vec{\phi})$ and the optimizer $\vec{y}^*$ for $b(\vec{\phi})$ are both unique and they satisfy $\vec{x}^* = \gamma \vec{y}^*$ for some $\gamma > 0$. Further, if the optimizer $\vec{x}^* \neq 0$, then $\vec{x}^*$ and $\vec{y}^*$ are the only vectors that satisfy the following conditions: there exist $\mu_m^i \geq 0$ such that $\sum\limits_{i=1}^N \mu_m^i = \phi_m$, $y_i^* = [\lambda_i - \sum\limits_{m=1}^M F_m^i \mu_m^i]^+$, $x_i^* = \gamma y_i^*$ for some $\gamma > 0$, $\sum\limits_{i=1}^N (x_i^*)^{\alpha+1} = 1$, and

$$\mu_m^i = 0 \text{ if } (x_i^*)^{\alpha} F_m^i < \max_k (x_k^*)^{\alpha} F_m^k.$$

Lemma 4 can be proved by showing that $b(\vec{\phi})$ is the dual problem of the optimization problem $a(\vec{\phi})$ with an appropriate change of variables. The variables μ_m^i of $b(\vec{\phi})$ are the Lagrange multipliers. Due to lack of space, we omit the proof here and the details are provided in our technical report [23].

We now show that, if the empirical measure process $\vec{s}(t)$ is linear, then the queue trajectory $\vec{x}(t)$ must also be linear, and it must solve $b(\vec{\phi})$ in Lemma 4. For ease of exposition, we start the time from $t = 0$ (instead of $t = -T$).

Lemma 5: Let $\vec{x}(0) = 0$ and $\vec{s}(t) = t\vec{\phi}$ for $t \geq 0$. Then the corresponding queue trajectory $\vec{x}(t)$ under the α-algorithm must satisfy the following:

(a) The queue trajectory is linear, i.e., for each i, $x_i(t) = \tilde{\mathbf{x}}_i t$ for some $\tilde{\mathbf{x}}_i \geq 0$.

(b) There must exist $\mu_m^i \geq 0$ such that $\sum\limits_{i=1}^N \mu_m^i = \phi_m$, and

$$\mu_m^i = 0 \text{ if } x_i^{\alpha}(t) F_m^i < \max_k x_k^{\alpha}(t) F_m^k \text{ for all } t.$$

In other words, the queue trajectory $\vec{x}(t)$ is consistent with the scheduling rule.

(c) $\vec{\tilde{\mathbf{x}}}$ is the unique minimizer of $b(\vec{\phi})$.

Proof: Let $\Omega(\vec{\phi}) = \left\{\vec{\lambda} \mid \lambda_i = \sum\limits_{m=1}^M \mu_m^i F_m^i, \sum\limits_{i=1}^N \mu_m^i = \phi_m, \mu_m^i \geq 0\right\}$. Note that $\Omega(\vec{\phi})$ would have been the capacity region (for stability) if the channel state distribution was $\vec{\phi}$.

Recall that (from (12))

$$\frac{dV(t)}{dt} = \left[\frac{\partial V_\alpha(\vec{x}(t))}{\partial \vec{x}}\right]^T \cdot \frac{d}{dt}\vec{x}(t)$$

$$= \left(\sum_{i=1}^{N} x_i^{\alpha+1}(t)\right)^{-\frac{\alpha}{\alpha+1}} \left[\sum_{i=1}^{N} x_i^{\alpha}(t)\lambda_i - \sum_{m=1}^{M} \phi_m \max_i x_i^{\alpha}(t) F_m^i\right].$$

If $\vec{\lambda} \in \Omega(\vec{\phi})$, we will have $\frac{dV(t)}{dt} < 0$ whenever $V(t) = V_\alpha(\vec{x}(t)) > 0$. Hence, starting from $\vec{x}(0) = 0$, we must have $V(t) = 0$ and $\vec{x}(t) = 0$ for all $t \geq 0$. Therefore, part (a) holds with $\tilde{\mathbf{x}}_i = 0$ for all i. Part (b) then trivially holds. Part (c) follows from Lemma 4 since the minimizer of $b(\vec{\phi})$ for this case is $\vec{y}^* = 0$.

On the other hand, if $\vec{\lambda} \notin \Omega(\vec{\phi})$, then for all $\vec{x}(t) \neq 0$, by setting $\tilde{x}_i(t) = \frac{x_i(t)}{[\sum\limits_{i=1}^{N} x_i^{\alpha+1}(t)]^{\frac{1}{\alpha+1}}}$, we have

$$\frac{dV(t)}{dt} = \sum_{i=1}^{N} \tilde{x}_i^{\alpha}(t)\lambda_i - \sum_{m=1}^{M} \phi_m \max_i \tilde{x}_i^{\alpha}(t) F_m^i$$

$$\text{and} \quad \left[\sum_{i=1}^{N} \tilde{x}_i^{\alpha+1}(t)\right]^{\frac{1}{\alpha+1}} = 1.$$

We thus have $\frac{dV(t)}{dt} \leq a(\vec{\phi})$ and $V(t) \leq ta(\vec{\phi})$. This shows that $ta(\vec{\phi})$ upper bounds the maximum growth of $V(t)$. On the other hand, let μ_m^i be the average fraction of time in $[0, t]$ that user i is picked and the channel state is m. Then $\sum\limits_{i=1}^{N} \mu_m^i = \phi_m$ for all m, and $x_i(t) \geq t[\lambda - \sum\limits_{m=1}^{M} \mu_m^i F_m^i]^+$. Hence,

$$V(t) = V_\alpha(\vec{x}(t)) \geq tb(\vec{\phi}).$$

However, by Lemma 4, $a(\vec{\phi}) = b(\vec{\phi})$. We thus have

$$V(t) = V_\alpha(\vec{x}(t)) = ta(\vec{\phi}) = tb(\vec{\phi}),$$

i.e., there is only one possible trajectory $V(t)$ given that $\vec{s}(t) = t\vec{\phi}$. Further, we have $V_\alpha(\frac{\vec{x}(t)}{t}) = b(\vec{\phi})$. i.e., $\frac{\vec{x}(t)}{t}$ optimizes $b(\vec{\phi})$. Since the optimizer of $b(\vec{\phi})$, denoted by $\vec{\tilde{\mathbf{x}}}$, is unique, we thus have $\vec{x}(t) = t\vec{\tilde{\mathbf{x}}}$. This shows parts (a) and (c). Part (b) follows from part (b) of Lemma 4. ■

Proposition 6: The minimum cost to overflow $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha)$ is equal to J_1.

Proof: Let $\vec{\phi}^*$, w^*, and $\vec{x}^*$ denote the solution to J_1. If we use $\vec{s}(t) = (t+T)\vec{\phi}^*, \quad t \geq -T$ as the underlying empirical measure process, and let the queue process start from $\vec{x}(-T) = 0$ where $T = 1/w^*$, then there is a linear trajectory according to Lemma 5, i.e.,

$$\vec{x}(t) = (t+T)\vec{\tilde{\mathbf{x}}}',$$

where $\vec{\tilde{\mathbf{x}}}'$ is the minimizer of $b(\vec{\phi}^*)$. Further, by the structure of J_1, $w^* \geq 0$, and thus

$$w^* = \max_{\vec{x} \geq 0} \left[\sum_{i=1}^{N} x_i^{\alpha}\lambda_i - \sum_{m=1}^{M} \phi_m \max_i x_i^{\alpha} F_m^i\right]$$
$$\text{subject to} \sum_{i=1}^{N} x_i^{\alpha+1} = 1.$$

The right hand side is equal to $a(\vec{\phi}^*)$, which is also equal to $b(\vec{\phi}^*)$. Hence,

$$V(0) = V_\alpha(T\vec{\tilde{\mathbf{x}}}') = Tb(\vec{\phi}^*) = \frac{1}{w^*}w^* = 1.$$

In other words, the linear empirical measure process $\vec{s}(t) = (t+T)\vec{\phi}^*, \quad t \geq -T$, indeed drives the queue from $\vec{x}(-\frac{1}{w^*}) = 0$ to overflow at $t = 0$. Hence, $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha) \leq TH(\vec{\phi}^*|\vec{p}) = \frac{1}{w^*}H(\vec{\phi}^*|\vec{p}) = J_1$. Then, using Lemma 3, $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha) \geq J_1$, the result then follows. ■

Hence, we conclude that the minimum-cost-to-overflow $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha)$ is attainable by a simple linear trajectory whose cost is given by $J_1 = \inf_{\vec{\phi}} \frac{H(\vec{\phi}|\vec{p})}{b(\vec{\phi})}$.

V. Asymptotical Optimality of α-algorithms

In this section, we return to the original overflow metric $h(\vec{x}) = \max_i x_i$ and we will establish that, in the limit as $\alpha \to \infty$, the α-algorithm asymptotically achieves the largest minimum-cost-to-overflow equal to I_{opt} given in Section III. We will use some of the results and notations from Section IV. In particular, to emphasize the dependence of $b(\vec{\phi})$ on α, we rewrite $b_\alpha(\vec{\phi}) = b(\vec{\phi})$ here:

$$\begin{aligned} b_\alpha(\vec{\phi}) \triangleq \min_{\vec{\phi}, \vec{x}} \quad & V_\alpha(\vec{x}) \\ \text{subject to} \quad & x_i = [\lambda_i - \sum_{m=1}^{M} \mu_m^i F_m^i]^+ \text{ for all } i \\ & \mu_m^i \geq 0, \sum_{i=1}^{N} \mu_m^i = \phi_m \text{ for all } m. \end{aligned} \tag{14}$$

In Section IV, we have shown that $\mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha) = \inf_{\vec{\phi}} \frac{H(\vec{\phi}|\vec{p})}{b_\alpha(\vec{\phi})}$. Earlier, in Section III, we showed that I_{opt} is an upper bound on the minimum-cost-to-overflow for all scheduling algorithms. We now show the following.

Theorem 7:

$$\lim_{\alpha \to \infty} \mathcal{F}(\Psi_{\alpha\text{-algo}}, \max) \geq I_{opt}.$$

Proof: First, it is easy to show that $\mathcal{F}(\Psi_{\alpha\text{-algo}}, \max) \geq \mathcal{F}(\Psi_{\alpha\text{-algo}}, V_\alpha)$. This is true because if a trajectory overflows according to the max-queue overflow metric, i.e., $\max_i x_i(t) = 1$, then it must have already overflowed according to the Lyapunov overflow metric since $\max_i x_i(t) = 1 \Rightarrow V_\alpha(\vec{x}(t)) \geq 1$.

Using Proposition 6, we then have

$$\mathcal{F}(\Psi_{\alpha\text{-algo}}, \max) \geq \inf_{\vec{\phi}} \frac{H(\vec{\phi}|\vec{p})}{b_\alpha(\vec{\phi})}.$$

We will now show that $\lim_{\alpha\to\infty} \inf_{\vec{\phi}} \frac{H(\vec{\phi}|\vec{p})}{b_\alpha(\vec{\phi})} = I_{opt} \triangleq \inf_{\vec{\phi}} \frac{H(\vec{\phi}|\vec{p})}{\acute{w}(\vec{\phi})}$, which then completes the proof.

Observe that $b_\alpha(\vec{\phi})$ in (14) and $\acute{w}(\vec{\phi})$ in (10) both have the same constraint set. The following inequality is easily established.

$$N^{\frac{1}{\alpha+1}} \max_i(x_i) \geq V_\alpha(\vec{x}) \geq \max_i(x_i) \text{ for all } \vec{x} \geq 0.$$

Hence,

$$N^{\frac{1}{\alpha+1}} \acute{w}(\vec{\phi}) \geq b_\alpha(\vec{\phi}) \geq \acute{w}(\vec{\phi}).$$

Note that this implies that $b_\alpha(\vec{\phi}) > 0 \Leftrightarrow \acute{w}(\vec{\phi}) > 0$. Let $\mathcal{Q} = \{\vec{\phi}$ such that $\acute{w}(\vec{\phi}) > 0\}$. It is sufficient to show

$$\lim_{\alpha \to \infty} \inf_{\vec{\phi} \in \mathcal{Q}} \frac{H(\vec{\phi}|\vec{p})}{b_\alpha(\vec{\phi})} = \inf_{\vec{\phi} \in \mathcal{Q}} \frac{H(\vec{\phi}|\vec{p})}{\acute{w}(\vec{\phi})}.$$

Now, for all $\vec{\phi}$ in $\mathcal{Q}$, the following holds

$$\frac{H(\vec{\phi}|\vec{p})}{\acute{w}(\vec{\phi})} \geq \frac{H(\vec{\phi}|\vec{p})}{b_\alpha(\vec{\phi})} \geq \frac{1}{N^{\frac{1}{\alpha+1}}} \frac{H(\vec{\phi}|\vec{p})}{\acute{w}(\vec{\phi})}.$$

Taking infimum across the inequalities over the set $\mathcal{Q}$, we get

$$\inf_{\vec{\phi} \in \mathcal{Q}} \frac{H(\vec{\phi}|\vec{p})}{\acute{w}(\vec{\phi})} \geq \inf_{\vec{\phi} \in \mathcal{Q}} \frac{H(\vec{\phi}|\vec{p})}{b_\alpha(\vec{\phi})} \geq \frac{1}{N^{\frac{1}{\alpha+1}}} \inf_{\vec{\phi} \in \mathcal{Q}} \frac{H(\vec{\phi}|\vec{p})}{\acute{w}(\vec{\phi})}.$$

Letting $\alpha \to \infty$, $N^{\frac{1}{\alpha+1}} \to 1$. The result of the Lemma then follows. ■

Combining Theorem 1 and Theorem 7, we conclude that the α-algorithm asymptotically achieves the largest possible value of the minimum-cost-to-overflow.

A. Systems with ON-OFF Channels

Consider the scenario where F_m^i can take either the value 0 or a positive constant C. This scenario corresponds to a wireless system with ON-OFF channels and the ON-rates for all users are the same. In this case, for any $\alpha > 0$,

$$x_i^\alpha F_m^i \lesseqqgtr \max_k x_k^\alpha F_m^k \Leftrightarrow x_i F_m^i \lesseqqgtr \max_k x_k F_m^k.$$

Hence, the α-algorithms (for any $\alpha \geq 1$) are equivalent to the max-weight algorithm (i.e., $\alpha = 1$). Using the result in this paper, we immediately reach the following corollary.

Corollary 8: For the above ON-OFF channel model, the max-weight scheduling algorithm achieves the largest minimum-cost-to-overflow I_{opt}.

VI. Conclusion

In this paper, we study wireless scheduling algorithms that can minimize the queue-overflow probability. Assuming that a sample-path large-deviation principle holds for the backlog process, we first establish a structural property of the minimum-cost-path-to-overflow for the class of α-algorithms. Specifically, when the overflow metric is appropriately modified, we show that the minimum-cost-to-overflow under the α-algorithm can be achieved by a simple linear path, and it can be written as the solution of a vector-optimization problem. Using this structural property, we then show that when α approaches infinity, the α-algorithm asymptotically achieves the largest value of the minimum-cost-to-overflow under all scheduling algorithms.

For future work, we plan to study conditions under which the sample-path large-deviation principle holds. We also plan to extend the results to more general network and channel models.

References

[1] X. Lin, N. B. Shroff, and R. Srikant, "A Tutorial on Cross-Layer Optimization in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, August 2006.

[2] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, December 1992.

[3] A. L. Stolyar, "Large Deviations of Queues under QOS Scheduling Algorithms," in *44th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2006.

[4] A. L. Stolyar and K. Ramanan, "Largest Weighted Delay First Scheduling: Large Deviations and Optimality," *Annals of Applied Probability*, vol. 11, no. 1, pp. 1–48, 2001.

[5] A. L. Stolyar, "Control of End-to-End Delay Tails in a Multiclass Networks: LWDF Discipline Optimality," *Annals of Applied Probability*, vol. 13, no. 3, pp. 1151–1206, 2003.

[6] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis: Queues, Communications, and Computing*. London: Chapman & Hall, 1995.

[7] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*, 2nd ed. New York: Springer-Verlag, 1998.

[8] A. I. Elwalid and D. Mitra, "Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High Speed Networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 329–343, June 1993.

[9] G. Kesidis, J. Walrand, and C.-S. Chang, "Effective Bandwidth for Multiclass Markov Fluid and other ATM Sources," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 424–428, Aug. 1993.

[10] C.-S. Chang, P. Heidelberger, S. Juneja, and P. Shahabuddin, "Effective bandwidth and fast simulation of ATM intree networks," *Performance Evaluation*, vol. 20, pp. 45–65, 1994.

[11] C.-S. Chang and J. A. Thomas, "Effective Bandwidth in High-Speed Digital Networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 1091–1114, Aug. 1995.

[12] F. P. Kelly, "Effective Bandwidth in Multiclass Queues," *Queueing Systems*, vol. 9, pp. 5–16, 1991.

[13] W. Whitt, "Tail Probabilities with Statistical Multiplexing and Effective Bandwidth for Multi-class Queues," *Telecommunication Syst.*, vol. 2, pp. 71–107, 1993.

[14] D. Wu and R. Negi, "Effective Capacity: A Wireless Link Model for Support of Quality of Service," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 630–643, July 2003.

[15] ——, "Downlink Scheduling in a Cellular Network for Quality of Service Assurance," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 5, pp. 1547–1557, September 2004.

[16] ——, "Utilizing Multiuser Diversity for Efficient Support of Quality of Service over a Fading Channel," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1198–1206, May 2005.

[17] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable Scheduling Policies for Fading Wireless Channels," *IEEE/ACM Transactions on Networking*, vol. 13, no. 2, pp. 411–424, April 2005.

[18] L. Ying, R. Srikant, A. Eryilmaz, and G. E. Dullerud, "A Large Deviations Analysis of Scheduling in Wireless Networks," *IEEE Transactions on Information Theory*, vol. 52, no. 11, November 2006.

[19] S. Shakkottai, "Modes of overflow, effective capacity and qos for wireless scheduling," in *Proceedings of IEEE International Symposium on Information Theory*, Yokohama, Japan, July 2003.

[20] A. Eryilmaz and R. Srikant, "Scheduling with Quality of Service Constraints over Rayleigh Fading Channels," in *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[21] S. Shakkottai, "Effective Capacity and QoS for Wireless Scheduling," *available at http://www.ece.utexas.edu/∼shakkott/pub.html*, 2004.

[22] X. Lin, "On Characterizing the Delay Performance of Wireless Scheduling Algorithms," in *44th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2006.

[23] V. J. Venkataramanan and X. Lin, "Structural Properties of LDP for Queue-Length Based Wireless Scheduling Algorithms," *Technical Report, Purdue University, http://min.ecn.purdue.edu/∼linx/papers.html*, 2007.