On the Stability Region of Linear-Memory Scheduling for Time Varying Channels

Mahdi Lotfinezhad, Ben Liang, Elvino S. Sousa
Department of Electrical and Computer Engineering
University of Toronto
{mlotfinezhad, liang}@comm.utoronto.ca, es.sousa@utoronto.ca

***Abstract*— Throughput optimal scheduling policies in general require the solution of a complex optimization problem. The past literature has shown that the complexity of this optimization problem can be greatly reduced, but at the expense of memory requirement that is exponential with the number of users. In this paper, we study the stability region of a class of linear-memory scheduling policies for time varying channels, and investigate how the channel memory impacts the supportable input rates. The set of scheduling policies in this paper covers a wide spectrum of resource allocation algorithms, which allows us to study policies with different complexity levels. In particular, we are able to model a class of low-complexity scheduling policies with linear memory, which are suitable for practical implementation.**

I. Introduction

Network layer capacity region is defined as the closure of the set of all input rates that can be *stably* supported by the network using any possible scheduling policy [1][2]. Obviously, an optimally designed network controller should achieve the highest possible throughput while ensuring network stability. Tassiulas and Ephremides in their seminal work [1] proposed a *throughput optimal* scheduling policy that stabilizes the network for any input rate that is within the network layer capacity region. Under time varying channel conditions, throughput optimal policies generally require the solution of a complex optimization problem involving access to information about the input queue backlog and the channel states. Recently, the authors of [3][4] showed that the complexity of throughput optimal schemes can be greatly reduced, but at the expense of memory requirement that is exponential with the number of users.

In this paper, we study the stability region of a general class of scheduling policies that have only linear memory requirement. Such scheduling policies are more scalable, but they generally do not achieve the throughput capacity. Hence we ask the question, "*How much sub-optimality in the network throughput is introduced by the reduced memory requirement?*" Clearly, the answer depends on the characteristics of the time-varying channel, as well as the complexity of the scheduling policy.

Following the general approaches studied in [5][3][4], the scheduling policy considered in this paper uses a randomized algorithm to generate a *candidate* resource allocation vector for the problem of throughput optimality in each timeslot. Then, by comparing the candidate and the previously used allocation vector, the policy utilizes an updating rule to select the most appropriate allocation vector for the current timeslot. The set of allocation-vector selection algorithms considered in this paper covers a wide spectrum which contains different algorithm with different complexity levels.

The use of randomized algorithms to reduce the complexity of throughput optimal scheduling first appeared in [5]. However, the scheduling policy in [5] is proposed in the context of non-time varying channels. Our focus in this paper is on the effect of channel memory on the stability region. Furthermore, we use a generalized version of the policy in [5]. In the context of time varying channels, other recent proposals that are based on the policy in [5] include [3][4]. Although they are throughput optimal, their memory requirement is exponential with the number of users, and thus, those proposals may not be amenable to practical implementation in large networks.

Our main contribution in this paper is to analytically characterize the stability region of the class of linear-memory policies for time varying channels. We show that the stability region is a scaled version of the network layer capacity region. Our analysis quantifies the scaling factor, and characterizes its dependence on the *channel memory* and the *computational efficiency* of the allocation-vector selection algorithm. In addition, our analysis provides insights into the extreme case of linear-complexity, linear-memory scheduling policies. Finally, aside from theoretical interests, our results also quantify the throughput gain of the linear-memory scheduling policy over arbitrarily random scheduling, even when only an approximate solution is available for the optimal allocation-vector selection problem.

II. System Model

We consider a one-hop wireless network consisting of N users each associated with a separate queue holding packets to be transmitted using a wireless link. We emphasize that our results hold for any one-hop wireless network. In particular, if users have multiple destinations, for each one, they create a separate queue. In that case, N represents the number of data flows, and corresponding to each source-destination pair, we consider a wireless link for data transmission.

The quality of the wireless links is measured by a state vector $\mathbf{s}(t) = (\mathbf{s_1(t)}, \dots, \mathbf{s_N(t)})$, where $\mathbf{s_i}$ is the channel condition of the i_{th} link. Let $\mathcal{S}$ represent the set of all possible channel states with finite cardinality. We assume the system is time-slotted, and the channels hold their state during a timeslot but may change from one timeslot to another. In every timeslot according to $\mathbf{s(t)}$, a controller picks a schedule vector $I \in \mathcal{I}$, which may include power levels as well as other physical layer parameters. The transmission rate vector at time t is $\mathbf{D(t)} =$

$(\mathbf{D_1}(\mathbf{t}), \ldots, \mathbf{D_N}(\mathbf{t}))$, where $\mathbf{D}(\mathbf{t})$ is uniquely characterized by the channel states and the selected schedule, and hence, we have $\mathbf{D}(\mathbf{t}) = \mathbf{D}(\mathbf{s}(\mathbf{t}), \mathbf{I}(\mathbf{t}))$. We assume that there are a finite number of ways to schedule users implying that $\mathcal{I}$ has finite cardinality. Therefore, the controller at any time t is confined to selecting a rate vector such that $\mathbf{D}(\mathbf{t}) \in \mathbf{\Gamma_{s(t)}}$ where $\mathbf{\Gamma_{s(t)}} = \{\mathbf{D}(\mathbf{s}(\mathbf{t}), I) | I \in \mathcal{I}\}$.

To avoid trivial complications, we assume that the arrival process is i.i.d. with mean rate vector $\mathbf{a}$, and is independent of the channel process. Furthermore, we assume that the second moment of the arrival process is finite. Assuming a timeslotted system, the transmission queue length vector at the users is defined by $\mathbf{X}(\mathbf{t}) = (\mathbf{X_1}(\mathbf{t}), \ldots, \mathbf{X_N}(\mathbf{t}))$. It evolves according to the following equation

$$\mathbf{X}(\mathbf{t+1}) = \mathbf{X}(\mathbf{t}) + \mathbf{A}(\mathbf{t}) - \mathbf{D}(\mathbf{t}) + \mathbf{U}(\mathbf{t}), \tag{1}$$

where $\mathbf{A}(\mathbf{t})$ and $\mathbf{D}(\mathbf{t})$ are the arrival and departure vectors, respectively. $\mathbf{U}(\mathbf{t})$ represents the wasted service vector with non-negative elements; the service is wasted when in a queue the number of packets waiting for transmission is less than the number that can be transmitted, i.e., when $\mathbf{X_i}(\mathbf{t}) < \mathbf{D_i}(\mathbf{t})$.

A. The Scheduling Policy

In [1] and recently under general conditions in [2], it has been shown that the capacity region is given by

$$\Gamma = \sum_{s \in \mathcal{S}} \pi(s) \text{ Convex-Hull}\{D(s, I) | I \in \mathcal{I}\},$$

where $\pi(s)$ is the channel state probability. It has been proved that a control policy that at any time t selects $\mathbf{D}^*(\mathbf{X}(\mathbf{t}), \mathbf{s}(\mathbf{t}))$ as the solution of the following optimization problem

$$\begin{aligned} \mathbf{D}^*(\mathbf{X}(\mathbf{t}), \mathbf{s}(\mathbf{t})) &= \max_{I \in \mathcal{I}} \mathbf{X}(\mathbf{t}) \mathbf{D}(\mathbf{s}(\mathbf{t}), I)^T \\ &= \max_{I \in \mathcal{I}} \sum_{i=1}^{N} \mathbf{X_i}(\mathbf{t}) \mathbf{D_i}(\mathbf{s}(\mathbf{t}), I), \end{aligned} \tag{2}$$

stabilizes the network for all input rates strictly inside Γ [1][2]. This policy, which is by definition throughput optimal, can be NP-complete due to physical layer interferences. In this work, we consider a policy based on a randomized algorithm to solve (2), which may be viewed as a generalized version of those studied in [5][6][3][4]. In the following, we first provide an abstract description of the scheduling policy and then provide the intuition behind the policy's operation.

1) Description of the Policy: In this paper, we assume that the network controller uses a randomized algorithm A to select a *random candidate* schedule from $\mathcal{I}$ at any given time t. In general, the distribution of the randomly selected schedule $\mathbf{I^r}(t)$ depends on $\mathbf{X}(\mathbf{t})$ and $\mathbf{s}(\mathbf{t})$. Let $\mu_{X,s}(.)$ denote such a distribution for I^r. The algorithm is, therefore, characterized by the collection of all distributions for different values of X and s, namely $\{\mu_{X,s}(.); s \in \mathcal{S}, X \in (\{0\} \cup \mathbb{Z}^+)^N\}$. Let $\mathcal{A}$ denote the collection of algorithms with the following properties:

P1 Assume $\|X_1 - X_2\| = C$, for some given constant C. Let $\|X_1\| \to \infty$, then $|\mu_{X_1,s}(I_i) - \mu_{X_2,s}(I_i)| \to 0$, for all $s \in \mathcal{S}$ and $I_i \in \mathcal{I}$.

P2 There exists a constant $1 > \zeta \geq 0$ such that, for all $X \in (\{0\} \cup \mathbb{Z}^+)^N$ and $s \in \mathcal{S}$, the following holds with probability $\delta > 0$:

$$XD(s, I^r)^T \geq (1 - \zeta) XD^*(X, s)^T. \tag{3}$$

P1 states that for $\mathrm{A} \in \mathcal{A}$, the distribution of I^r would be almost the same when two queue length vectors are *close* enough. P2 states that the selected schedule with probability $\delta > 0$ is in ζ neighborhood of the optimal solution. This property is a generalized version of the one in [5][6].

Using algorithm A, the network controller chooses the schedule at any time t, $\mathbf{I}(\mathbf{t})$, as follows:

- $\mathbf{I}(\mathbf{0})$ is selected randomly according to $\mu_{\mathbf{X}(\mathbf{0}), \mathbf{s}(\mathbf{0})}(.)$.
- For $t > 0$, $\mathbf{I}(\mathbf{t})$ is determined by the following steps.
  - First, $\mathbf{I^r}(\mathbf{t})$ is selected randomly following the distribution $\mu_{\mathbf{X}(\mathbf{t}), \mathbf{s}(\mathbf{t})}(.)$
  - Then, we have

$$\mathbf{I}(\mathbf{t}) = \begin{cases} \mathbf{I^r}(\mathbf{t}) & \text{w.p. } f(\varphi(t)) \\ \mathbf{I}(\mathbf{t-1}) & \text{otherwise} \end{cases},$$

where

$$\varphi(t) = \frac{\mathbf{X}(\mathbf{t})(\mathbf{D^r}(\mathbf{t}) - \mathbf{D}'(\mathbf{t-1}))^T}{\max(\mathbf{X}(\mathbf{t})\mathbf{D^r}(\mathbf{t})^T, \mathbf{X}(\mathbf{t})\mathbf{D}'(\mathbf{t-1})^T) + \alpha \|\mathbf{X}(\mathbf{t})\|}.$$

In the above, $\mathbf{D^r}(\mathbf{t}) = \mathbf{D}(\mathbf{s}(\mathbf{t}), \mathbf{I^r}(\mathbf{t}))$, $\mathbf{D}'(\mathbf{t}) = \mathbf{D}(\mathbf{s}(\mathbf{t+1}), \mathbf{I}(\mathbf{t}))$, and α is an arbitrarily small positive constant. $\mathbf{D}'(\mathbf{t})$ is simply the rate at the next timeslot assuming that the controller keeps using the current schedule $\mathbf{I}(\mathbf{t})$ at time $t+1$. The continuous function $f\colon (-1, 1) \to [0, 1]$ has the property that $f(\varphi) = 1$, for $\varphi \geq \rho > 0$, and $f(\varphi) = 0$, for $\varphi \leq -\rho$, where ρ can be chosen arbitrarily small. The introduction of $f(\varphi)$ in the decision making process allows the policy to take soft decisions when comparing two different schedules, i.e., the policy can probabilistically choose either of the schedules according to the value of $\varphi(t)$. A close look at the policy description shows that only $\mathbf{I}(\mathbf{t-1})$ is required to update $\mathbf{I}(\mathbf{t})$ in each timeslot. Therefore, the scheduling policy needs to keep track of only the previously used schedule, which results in a memory requirement that is linear with the number of users.

2) Intuitive Explanation of the Policy: As a practical example, consider the downlink of a CDMA network. Since power budget is limited in CDMA networks, the scheduler in each base station maintains the sum of the allocated power levels less than or equal to a system dependent value. In this example, the schedule vector $\mathbf{I}(\mathbf{t})$ is the vector of all possible power levels. If the scheduler uses the policy given by (2), it should find a power allocation vector $\mathbf{I}(\mathbf{t})$ that maximizes the sum-product of backlog-rate. This optimization problem is non-convex due to the physical layer interferences, and in general, can be NP-complete. Thus, we assume the scheduler has access to an algorithm that returns a power allocation vector that is in ζ neighborhood of the optimal solution with probability δ. Assuming different values for δ and ζ allows us to consider algorithms with different complexity levels.

In contrast to the throughput optimal policy given by (2), the scheduling policy in this paper uses the randomized algorithm to find a candidate power allocation vector $\mathbf{I^r(t)}$. The policy then compares $\mathbf{I^r(t)}$ with the previously used vector $\mathbf{I(t-1)}$. For the comparison, it evaluates $\varphi(t)$ as the relative difference of backlog-rate sum-products corresponding to $\mathbf{I^r(t)}$ and $(\mathbf{t-1})$. The computation of φ is based on the current channel state $\mathbf{s(t)}$. A positive $\varphi(t)$ implies that the choice should favor the selection of $\mathbf{I^r(t)}$. However, any selected schedule affects future decisions, and due to channel variations may negatively impact throughput in the subsequent timeslots. Therefore, the value of $\varphi(t)$, especially in the neighborhood of zero, may not be the only measure to select $\mathbf{I^r(t)}$ or $\mathbf{I(t-1)}$. Thus, by introducing $f(\varphi)$, we adopt a probabilistic approach. In this paper, we study the stability region assuming that the function $f(\varphi)$ is given, and leave finding the optimal $f(\varphi)$ as an interesting open problem for future research. It is easy to see that there exist instances of low-complexity algorithms that satisfy P1 and P2, and have linear complexity in N [5].

B. Channel State Process

We assume that the channel state process satisfies the following properties.

A1 The channel state process is stationary and ergodic, and for any given ϵ_1 there exists K_{ϵ_1} such that for $K > K_{\epsilon_1}$ regardless of the initial state $\mathbf{Y(t)} = (\mathbf{X(t)}, \mathbf{I(t)}, \mathbf{s(t)})$, we have

$$|\pi(s) - \mathbb{E}[\frac{1}{K}\sum_{k=0}^{K-1} \mathbf{1}_{\mathbf{s}(t+k)=s}|\mathbf{Y_t}]| < \epsilon_1. \tag{4}$$

A2 We show in the next section that the network stability depends on how the policy of Section II-A.1 selects schedules when the backlog vector is frozen at $\mathbf{X(t)}$ after a particular time t.[1] We use the notation $\mathbb{E}_{\mathbf{X_t}}[.]$ and $p_{\mathbf{X_t}}(.)$ to represent the expectation and distribution of any random variable that is determined by the channel process and $\mathbf{X}(t)$, given the hypothesis that $\mathbf{X}(t^{'}) = \mathbf{X}(t)$ for $t^{'} > t$. We assume that $\mathbb{E}_{\mathbf{X_t}}[g(.)]$ is *uniformly* upper bounded in t by its $\limsup$ over t and other parameters that define the random variable $g(.)$. Note that if the channel process is Markovian (or n-generalized Markovian) we do not need this assumption.

III. Network Stability

In this section, we study the stability region of the network under any instance of the policies given in Section II-A.1. By the stability region, we mean a region that provides a sufficient condition for an input rate to be stably supported by a given control policy. Define $\Sigma_{\mathbf{X_t}}^K$ as

$$\Sigma_{\mathbf{X_t}}^K = \frac{\sum_{m=0}^{K-2} \mathbb{E}_{\mathbf{X_t}}[\mathbf{X_t}(\mathbf{D_{t+m}} - (1-\rho)\mathbf{D}^{'}_{\mathbf{t+m}})^T|\mathbf{Y_t}]}{K\mathbb{E}[\mathbf{X_t}\mathbf{D}^*(\mathbf{X_t},\mathbf{s})|\mathbf{X_t}]},$$

where $\mathbb{E}[\mathbf{X_t}\mathbf{D}^*(\mathbf{X_t},\mathbf{s})|\mathbf{X_t}]$ is computed over the steady state distribution of the channel process. In the above, since, as defined in Section II-A.1, $\mathbf{D}^{'}_{\mathbf{t}}$ is the rate vector corresponding to the allocation vector in the current time slot, $\mathbf{I(t)}$, and the channel state in the next timeslot, $\mathbf{s(t+1)}$, $\mathbf{D_{t+m}} - (1-\rho)\mathbf{D}^{'}_{\mathbf{t+m}}$ approximately shows the changes in the rate vector $\mathbf{D_{t+m}}$ due to channel variations. Note that the sequence of $\mathbf{D_i}$'s not only depends on the particular realization of channel states but also on the randomized algorithm that finds the candidate schedules. Therefore, when K is large, $\Sigma_{\mathbf{X_t}}^K$ measures the relative change of backlog-rate product due to channel variations over a long horizon while implicitly embedding the effects of the randomized algorithm. It is important to note that in the definition of $\Sigma_{\mathbf{X_t}}^K$, the backlog vector is kept at $\mathbf{X_t}$. This implies that $\Sigma_{\mathbf{X_t}}^K$ is not affected by the arrival process. Let Σ^∞ be the $\limsup$ of $\Sigma_{\mathbf{X_t}}^K$, i.e.,

$$\Sigma^\infty = \limsup_{t\to\infty, K\to\infty, x\to\infty} \sup_{\|\mathbf{X_t}\|=x} \Sigma_{\mathbf{X_t}}^K. \tag{5}$$

Σ^∞ measures the limiting value of the relative rate change due to the channel variations. Our results indicate that Σ^∞ directly affects the supportable input rates. The main result of this paper is stated in the following theorem.

Theorem 1: If the mean arrival rate vector, $\mathbf{a}$, lies strictly inside $(\theta-\epsilon)\Gamma$, under the control policy of Section II-A.1 the system described in Section II is stable in the mean, i.e.,

$$\limsup_{T\to\infty} \frac{1}{T+1}\sum_{l=0}^{T} \mathbb{E}[\|\mathbf{X}_l\|] < \infty, \tag{6}$$

where $\theta = 1 - \zeta^{'} - \frac{1-\delta}{\delta}\Sigma^\infty$, $\zeta^{'} = (1-(1-\rho)(1-\zeta))$, and $\epsilon = c\rho\alpha\delta^{-1}$ for some $c > 0$.

The detailed proof of the theorem is provided in [7].

A. Discussion

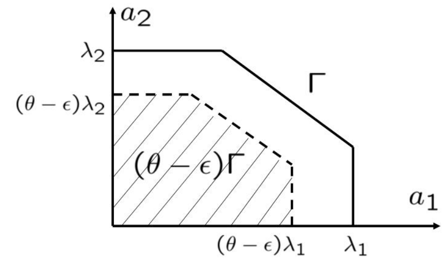

Fig. 1. Illustration of the stability region scaling.

As the above theorem suggests, the guaranteed stability region is a scaled version of the capacity region Γ. The stability region scaling is illustrated in Fig 1. The theorem, moreover, shows that the scaling factor, $\theta - \epsilon$, depends on the limiting behavior of the algorithm when the queue length vector is frozen at some specific vector. Note that since α and ρ are very small positive numbers, and α can be chosen arbitrarily, ϵ is negligible compared to θ.

This theorem also states that the stability region is affected at most linearly with rate changes due to channel variations. This can be easily seen from the definition of Σ^∞. In particular, when the maximum rate change is $r\%$, we have $\Sigma^\infty \leq \frac{r}{100}$, and therefore, θ changes at most linearly with r. One important observation is that as $r \to 0$, i.e., as channel states become more

[1] This means that after time t the backlog vector does not get updated, however, the channel states still evolve, and the control policy selects schedules.

correlated, $\theta \to 1-\zeta^{'}$. This implies that θ becomes independent of δ, and hence, this theorem states how the channel memory helps reduce the uncertainty of the randomized algorithm A in selecting a candidate schedule satisfying (3). Therefore, this general scheduling policy has the ability to take advantage of the channel memory to compensate for the inefficiencies in the randomized algorithm. The result is a significant expansion of the stability region when channel states are highly correlated. For example, consider the linear-complexity algorithm given in [5] with $\zeta = 0$. Assume the extreme case that the channel states do not change with time. It can been seen that as $\rho \to 0$, $\theta \to 1$. Noting the fact that throughput optimality can be achieved when $\theta - \epsilon = 1$, we conclude that if the channel states are fixed, using simple linear-complexity algorithms is sufficient to attain throughput optimality arbitrarily closely, which is reminiscent of the results in [5].

B. A Simple Example

To gain more insights, we apply the theorem to a network consisting of two users. Suppose there are two channel states s_1 and s_2, and three different schedules I_1, I_2, and I_3. For simplicity, we assume that the channel process is Markovian with symmetric transition probability t_p. Table I shows the rate vector $\mathbf{D}$ corresponding to each channel state and each selected schedule. We consider a very simple algorithm that selects one of the I_i's with probability $\frac{1}{3}$. Consequently, for this randomized algorithm we have $\delta = \frac{1}{3}$ and $\zeta = 0$. To compute $\varphi(t)$, we assume $\alpha = 0.001$, $\rho = 0.01$, and $f(\varphi) = 0.5 + \frac{\varphi}{2\rho}$ for $|\varphi| < \rho$.

Fig. 2 shows the numerically calculated $\theta - \epsilon$ as a function of t_p. As t_p decreases, the channels states become more correlated. We have also shown the performance of two other related control policies. The first policy CP1 was initially studied in [3][5]. For all states $\mathbf{s}$, CP1 records the last schedule that has been used at the most recent instance of $\mathbf{s}$, and therefore, it has to maintain a large table with an exponential size in terms of the number of users. This policy at any time uses the randomized algorithm of Section II-A.1 to select a candidate schedule when the channel is at $\mathbf{s(t)}$, except that it compares this schedule with the schedule used at the last instance of $s = \mathbf{s(t)}$. The control policy in Section II-A.1, however, compares with the very immediate schedule used in the previous timeslot. Results in [3][5] indicate that the stability region of CP1 equals to Γ. In contrast, the second policy, CP2, simply chooses a schedule at random, and does not do any comparisons at all. Using a simple drift analysis, it can be shown that the stability-region scaling factor of CP2 equals $\delta(1-\zeta)$.

This figure indicates that the linear-memory policy of Section II-A.1 significantly outperforms CP2; even in the extreme case when $t_p \to 1$. On the other hand, as t_p decreases, the stability region of this policy quickly expands to Γ. Therefore, when the channels states are highly correlated, the linear-memory policy stabilizes the network under most allowable input rates.

IV. Conclusion

In this paper, we have studied the stability region of a general class of linear-memory scheduling policies for time varying channels. We have shown that this class of policies can be used to model scheduling algorithms with different complexity levels whose complexity can be as low as linear. We have proved that if the mean arrival rate vector lies strictly inside a scaled version of the network-layer capacity region, then the studied policies are able to stabilize the network. Furthermore, we have quantified the scaling factor of the stability region based on the limiting behavior of rate changes due to channel variations and the inefficiencies inherent in the scheduling policies. In addition, our results indicate how channel memory helps reduce the uncertainty of the scheduling policy in selecting a suitable candidate schedule. Finally, we have shown that our results in the extreme case of non-time varying channels reduce to the ones in [5], where it is shown that linear complexity algorithms can be used to attain throughput optimality.

TABLE I

Rate vectors corresponding to each channel state and schedule

	I_1	I_2	I_3
s_1	(1, 0)	(0, 0)	(0.5, 0.25)
s_2	(0, 0)	(0, 1)	(0.25, 0.5)

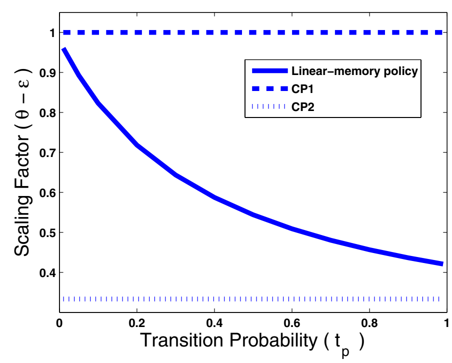

Fig. 2. Scaling factor of the capacity region as a function of the transition probability (ρ=0.01, α=0.001).

References

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

[2] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[3] A. Eryilmaz, R. Srikant, and J. Perkins, "Stable scheduling policies for fading wireless channels," *IEEE/ACM Trans. Networking*, vol. 13, no. 2, pp. 411–424, Apr. 2005.

[4] P. Chaporkar and S. Sarkar, "Stable scheduling policies for maximizing throughput in generalized constrained queueing," in *Proc. IEEE INFOCOM'06*, apr 2006.

[5] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," *Proc. IEEE INFOCOM'98*, vol. 2, 1998.

[6] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," *Proc. IEEE INFOCOM'05*, vol. 3, pp. 1804–1814, Mar. 2005.

[7] M. Lotfinezhad, B. Liang, and E. Sousa, "On the stability region of linear-memory low-complexity scheduling for time varying channels," manuscript submitted for journal publication, 2007.