Opportunistic Scheduling: Generalizations to Include Multiple Constraints, Multiple Interfaces, and Short Term Fairness

Sunil Suresh Kulkarni, Catherine Rosenberg

School of Electrical and Computer Engineering

Purdue University, West Lafayette, IN 47907-2035, USA

Email: sunilkul,cath@ecn.purdue.edu

Abstract

We consider several scheduling problems for packet based systems with time-varying channel conditions. Designing scheduling mechanisms that take advantage of time-varying channel conditions, which are different for different users, is necessary to improve system performance; however this has to be done in a way that provides some level of fairness among the users. Such scheduling mechanisms are termed opportunistic. We generalize the opportunistic scheduling mechanisms in the literature on three fronts. First, we formulate and solve an opportunistic scheduling problem with multiple general long term QoS constraints and a general system objective function. The solution of this opportunistic scheduling problem is an index policy. Then, we generalize this problem to include multiple interface systems in which several users can be served simultaneously. Apart from the long term QoS constraints specified by each user, multiple interface systems are constrained with other physical limitations imposed by the system. We show that the structure of the optimal opportunistic scheduling policy is carried over to the problem with general constraints and multiple interfaces. We also study the stability of the multiple interface systems and propose a throughput optimal scheduling rule for such systems. We then formulate an opportunistic scheduling problem with short term processor sharing fairness constraints as an optimization problem where fairness is guaranteed over a finite time window. In its most general form, this problem cannot be solved analytically. Hence observing the form of the optimal policies for special cases, we propose a heuristic scheduling policy. We illustrate the effectiveness of the policies via simulation.

I. INTRODUCTION

Wireless channels, in contrast to their wireline counterparts have time-varying location-dependent characteristics and different wireless users experience different channel conditions at a given time. These channels are affected by user shadowing, interference and path losses due to changing environments and due to user mobility. In CDMA based systems, wireless channels are affected by co-channel interference due to other users. Also, in the case of satellite systems, the channel conditions vary due to the weather conditions and satellite movements. In prior works [3, 7], it has been argued that the variations in the channel conditions should be exploited to increase the system throughput. The basic idea behind exploiting the channel variations is to schedule a user having the best channel condition at a given time. Such scheduling mechanisms are called *opportunistic*. If the service requirements of all the users are flexible, such opportunistic scheduling mechanisms can result in reduced interference, higher spectrum utilization, and increased system throughput. Use of opportunistic scheduling schemes can give throughput gain of more than 100% with respect to non-opportunistic scheduling such as round robin. The effectiveness of opportunistic scheduling has been accepted by the research community and such methods have been incorporated in the design of new generations of wireless systems such CDMA-HDR (IS-856). This is an example of a high data rate system that takes advantage of time-varying channel conditions through the use of an opportunistic scheduling mechanism.

The problem of packet scheduling for single interface systems with time-varying channel conditions can be illustrated as follows (see Figure 1). Consider a base station with fixed transmission power and a packet based downlink scheduling mechanism. The wireless channel for each user differs depending on the location, the surrounding environment, and mobility. Assume that each user reports its downlink channel condition to the base station in a periodic fashion. Thus the base station knows the current channel condition and hence the data rate it can offer to each user on the downlink channel if only that user is served at the given time. After finishing a packet transmission, the base station must choose the next (single) user to whom it will send the next packet (that can be done by inspecting the current potential transmission rates of the different active users).

For single interface systems, we formulate a multiple constraint opportunistic scheduling problem with long term QoS constraints (*i.e.*, constraints that are based on long term averages). *We show that the solution is an index policy in Section III-A.* We also discuss relevant system considerations in the same section.

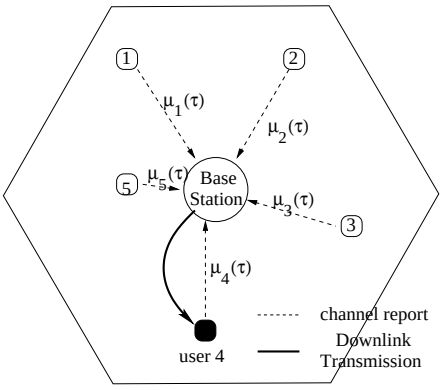

Fig. 1. Single interface opportunistic scheduling

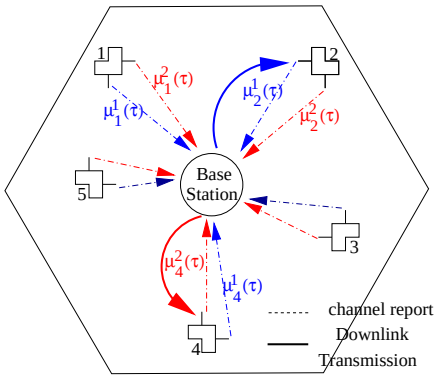

Fig. 2. Multiple interface opportunistic scheduling

There are systems of interest which cannot be modeled in the above framework. Among them are the *multiple interface systems* where the user channel conditions are time-varying over all the interfaces and the base station has the freedom to serve several users on several interfaces simultaneously. For example, in multi-carrier CDMA systems every user has a time-varying channel on each carrier. In OFDM systems, fading and shadowing are different for each carrier. In multiple beam satellite systems, a satellite can transmit simultaneously to different ground terminals over different channels. Depending on the specifics of the system, the number of interfaces can vary from two to more than thousand.

To describe the scheduling problem in such systems, consider Figure 2. All the transmissions on different interfaces start and end at the same time. Each user has different channel conditions over different interfaces. Assume that each user reports its channel condition on every interface to a single base station. The base station must schedule several users for transmission over multiple interfaces in a way that satisfies the physical constraints. The physical constraints can differ from system to system taking into account the different technologies used at the base station, but these physical constraints must be satisfied at all times. We do not assume that the different interfaces use the same technology, but we do assume that the base station has complete knowledge of the current channel state for each user on each interface. This excludes the case in which each interface is managed by a different base station. It is not necessary that all users have all the interfaces. If a particular user does not support a particular kind of interface, the data transmission rate for that user over that kind of interface is always assumed to be zero. Thus, without loss of generality we assume that all users support all interfaces. These kind of systems have received much less attention so far. *In Section III-B we formulate an opportunistic scheduling problem over multiple interfaces and show that the*

solution is again an index policy. We also give a throughput optimal scheduling rule that guarantees stable user queues for such systems in Section III-C.

Opportunistic scheduling mechanisms have to take into account some notion of fairness. Most of the studies in the literature so far have focused on long term fairness constraints. However wireless channels are correlated and non-stationary; users in deep fades experience a bad channel for prolonged periods of time. Hence long term policies may lead to long starvation period for such users. A good scheduling policy must guarantee fair share of the network resources to each user over some finite time window. There is a need for a policy which takes into consideration the short term requirements of the users because usually networking protocols have some timers associated with them. All these timers at different protocol layers interact with each other in an unpredictable manner. An expiration of a timer is a *bad* event for an end-to-end connection. Such an event is usually interpreted as an indication of congestion or loss of connectivity. Hence we would like to give a strict guarantee on the maximum starvation period, *i.e.*, the maximum period between two successive service offerings to an active user. Usually this guarantee will be the same for all the users. If the minimum possible data rate for a user is strictly greater than zero then a guarantee on the maximum starvation period would automatically correspond to a minimum data rate for that user. *We define, in Section IV-A, a short term fairness constraint and formulate an opportunistic scheduling problem with this constraint.* As this problem is very complex to solve under general conditions, *we consider some simple cases in Section IV-B.* With the insights from the optimal opportunistic scheduling policies for these special cases *we design a heuristic policy for the general opportunistic scheduling problem in Section IV-C.*

This paper is organized in the following way. In Section II, we discuss previous relevant work in this area. In Section III we formulate and study opportunistic scheduling problems with long term fairness constraints. In Section IV we formulate and study an opportunistic scheduling problem with short term fairness constraints. In both these sections we present numerical results. We conclude in Section V. The results in this paper have been presented in two conferences ([5] and [6]).

II. Previous work

The problem of exploiting the channel state variations to increase the throughput of wireless systems has been in focus in recent years. The typical representative term used to describe systems using opportunistic scheduling is

multi-user diversity. Opportunistic scheduling schemes have been studied in the literature mostly for single interface systems, with a single form of long term fairness constraints.

Multi-user diversity has a similar effect to that of a space or a time diversity, enabling some improvement in the system performance. This multi-user diversity has been described from an information theoretic viewpoint in [15] and the references contained therein. In [3] the authors present a similar opportunistic scheduling scheme for the Qualcomm/HDR system. These schemes provide asymptotic proportional fairness among the users. In [16] the authors suggest to increase or even introduce random variations in the channel conditions using multiple dumb antennas so that the multi-user diversity can be fully exploited. They analyze such a system and give the asymptotic bounds for the total channel capacity. But the simulations with practical system parameters do not confirm these results due to various practical limitations (for more details please refer to [4]). The main drawback of all these approaches is that individual QoS requirements cannot be taken into account in designing the scheduling policy.

In [10, 12] the authors extend wireline scheduling policies to wireless networks and present wireless fair scheduling policies which give short term and long term fairness bounds. While this approach provides fairness guarantees, it assumes that the channel quality is either good or bad. In [8] the authors present a scheduler called WCFQ (Wireless Credit-based Fair Queuing) to provide (only) statistical fairness bounds on the fraction of the processor share received by each user. Their approach is based on CBFQ (Credit Based Fair Queuing), a scheduler for wired systems. WCFQ trades off fairness for throughput to exploit the channel time variations by mapping channel conditions into a cost function. In [13] the authors present and study a practical scheme to exploit channel time variations in 802.11 based ad-hoc networks. While maintaining the same level of fairness, they provide a modification of the 802.11 protocol to obtain a better throughput than that of the original 802.11 protocol.

In [1, 14], the authors study throughput optimal scheduling rules. If there exists any scheduling rule which can guarantee stable queues then a throughput optimal scheduling rule also guarantees stable queues. Thus, throughput optimal scheduling rules have the largest system stability region in the space defined by the average incoming rates. Throughput optimal scheduling rules in time-invariant case are studied in [2]. In [11], the authors study a specific case of multi-beam satellite systems, propose a throughput optimal scheduling rule, and extend results to wireless networks.

In [7], the authors introduce three different QoS constraints, which they call Resource Sharing Fairness Constraint, Performance Based Fairness Constraint, and Minimum Performance Fairness Constraint. They provide optimal op-

portunistic scheduling mechanism for these special constraints. They also present one way of achieving *soft* short term fairness. In [14] the authors present a technique to extend the throughput optimal scheduling rules to guarantee minimum data rate constraints. Most of these results except [9] consider only the single interface case. Specifically, it is inherently assumed that only one user will be scheduled at a given time. In [9] the authors consider a throughput maximization problem with deterministic and probabilistic GPS-like (Generalized Processor Share) fairness constraints for multiple interface systems. But their approach is limited to a specific objective and specific constraints.

Our work not only generalizes the previous studies on many fronts but also unifies many of the results found in the literature and puts all of them into a single framework.

III. Opportunistic Scheduling with Long Term QoS Constraints

In this section we study opportunistic scheduling problems with long term QoS constraints.

A. Opportunistic Scheduling over a Single Interface with Multiple Constraints

To formulate our scheduling problem for a time-varying channel over a single interface with multiple QoS constraints, we make the following simplifying assumptions.

1) We assume that the system operates on a timeslot by timeslot basis. The width of each timeslot is fixed and the channel conditions do not vary during a timeslot. In real systems, the physical frame size can be selected from a finite discrete set. Hence depending on the current data rate, the transmission of a physical frame can take more than one timeslot. However in our model we assume that the physical frame transmission size can be varied according to the transmission rate so that transmissions begin and end exactly at timeslot boundaries. This is also assumed in [1, 7, 11, 14–16].
2) In real systems, the users send the necessary channel state information to the base station in a periodic fashion. Hence we assume that the base station knows at the beginning of each timeslot the exact channel state and the exact data rate with which it can transmit to each user.
3) We assume that at most one user can be served in a timeslot.
4) We also assume that all the users are greedy (i.e., each user always has data to receive on the downlink).

Assumptions 1, 2 and 4 are valid throughout the paper while assumption 3 is not always valid.

Constraints and Objectives: The QoS requirements for different users can be different and each user can potentially specify its own requirements. In general, these requirements can be grouped into short term and long term constraints. In this section we only consider long term requirements. The two most important long term QoS constraints are:

- Processor sharing constraint: User i specifies a weight ϕ_i and expects to get at least a fraction of the server time equal to ϕ_i. Thus this fairness constraint can be viewed as a fairness criterion similar to the generalized processor sharing. For a stable system, obviously we need $\sum_i \phi_i \leq 1$.
- Data rate constraint: A user asks for a minimum data rate guarantee to sustain its applications. This is a more appealing QoS criterion from the user standpoint, as most of the non-elastic applications need some minimum data rate. However a user might request a minimum data rate but may experience very poor channel conditions. Hence, it is not easy to specify the feasibility of the system under general minimum performance constraints because we do not assume any knowledge of the channel conditions.

We have mentioned only few possible fairness or QoS requirements. But potentially there can be many more and combinations of those constraints can also be specified. For example, a wireless node may specify two constraints, one for maximum power consumption and the other for minimum data rate requirement. The main constituent of the energy consumption is the energy consumed in radio electronics. To satisfy the maximum power consumption constraint, an upper limit on the processor time share can be calculated. Thus rather than having a lower bound on the processor share (as in GPS), there could potentially be an upper bound on the processor share to reduce the power consumption. By putting an upper limit on the processor time share a mobile device can guarantee a maximum power drain.

Notation: We start by introducing the notation and then state the general constraints associated with all the users. We use similar terminology and notation as in reference [7].

- $\mathcal{N}$: This denotes the set of users, usually indexed by i. The users will be indexed from 1 to N.
- $\mu_i(t)$: This denotes the data rate for user i in timeslot t. Thus $\vec{\mu}(t) = [\mu_1(t), \cdots, \mu_N(t)]$ denotes the vector of the data rates for all users at time t.
- $\mathcal{K}_i$: This denotes the set of constraints for user i. The constraints are usually indexed by j.
- f_i, g_i^j: f_i denotes the system utility function, and g_i^j denotes the j^{th} constraint function associated with user i. We assume that the f_i and g_i^j are convex functions in their arguments.

- $Q(\vec{\mu}(t))$: This denotes a scheduling policy to select a user to serve in timeslot t, given $\vec{\mu}(t)$.

We denote the indicator function by the letter I, thus $I_{Q(\vec{\mu(t)})=i}$ is 1 if in timeslot t user i is selected for service by policy Q, otherwise zero. We assume that each user is only interested in long term QoS constraints. Omitting the timeslot variable t by some abuse of notation, for a user i the j^{th} long term QoS requirement constraint is stated as follows: $\forall i \in \mathcal{N}, j \in \mathcal{K}_i, \;\; E\{g_i^j(\mu_i) I_{Q(\vec{\mu})=i}\} \geq G_i^j$

The above constraint is a long term constraint as the QoS requirement is to be guaranteed in *expectation*. Thus assuming ergodicity over a long period of time, the QoS constraint would be satisfied on an average. For the processor sharing constraint, the functions g_i^j can be taken as unit functions, $g_i^j(\mu_i(t)) = 1$. Thus the processor sharing constraint can be written as $\forall i, \;\; E\{I_{Q(\vec{\mu})=i}\} \geq \phi_i$. In words, this constraint can be phrased as, on the long term a user i should get a base station time share greater than or equal to ϕ_i. Similarly for the data rate constraint, g_i^j can be taken as $g_i^j(\mu_i(t)) = \mu_i(t)$ and hence the data rate constraint can be written as $\forall i, \;\; E\{\mu_i I_{Q(\vec{\mu})=i}\} \geq R_i$ (R_i being the minimum data rate requested by user i).

We define a general system objective as follows: $max_Q \sum_{i \in \mathcal{N}} E\{f_i(\mu_i(t)) I_{Q(\vec{\mu}(t))=i}\}$.

The above objective is also a long term objective in which the system utility is to be maximized in an expected sense. Thus assuming ergodicity, over a long period of time the average system utility should be maximized. Usually the objective of the system is to maximize the total system throughput. In that case the utility functions f_i can be taken as $f_i(\mu_i(t)) = \mu_i(t)$. Intuitively this would force the optimal policy to choose a user having a better channel (higher data rate) to maximize the system throughput.

Problem Formulation and Solution: The single interface scheduling problem with multiple constraints can be formulated as an optimization problem as follows.

P 1: $$max_Q \sum_{i \in \mathcal{N}} E\{f_i(\mu_i) I_{Q(\vec{\mu})=i}\} \quad such\ that \quad ; \forall i \in \mathcal{N}, \; \forall j \in \mathcal{K}_i \quad E\{g_i^j(\mu_i) I_{Q(\vec{\mu})=i}\} \geq G_i^j$$

Theorem 1: The solution Q^* of the above single interface constrained opportunistic scheduling problem, if one exists, is of the following form:

$$\exists \lambda_i^j \geq 0 \;\; s.t. \;\; Q^* = argmax_i \{f_i(\mu_i) + \sum_{j=1}^{\mathcal{K}_i} \lambda_i^j g_i^j(\mu_i)\} \quad \text{where} \quad E\{g_i^j(\mu_i) I_{Q(\vec{\mu})=i}\} > G_i^j \Rightarrow \lambda_i^j = 0 \qquad (1)$$

Proof: The proof follows directly by formulating the Lagrangian and follows the same steps as in the proof of the Theorem 2 in Section III-B of this paper. The details are omitted for the sake of space. ■

The constants λ's are the Karush-Kuhn-Tucker (KKT) multipliers and depend on the multidimensional distribution of the $\mu_i(t)$'s. The optimal policy is always of an argmax type for the type of constraints and objectives we have defined. The arguments for the argmax are the weighted and shifted sums of functions of the current data rate for each user. If all the users specify only the processor sharing constraint, the optimal policy becomes $argmax_i\{\mu_i + \lambda_i\}$. Thus the optimal policy (for processor sharing constraints) adds a bias equal to the KKT multipliers to the data rate values. If all the users specify only a minimum data rate constraint, the optimal policy becomes $argmax_i\{\mu_i(1+\lambda_i)\}$. Thus the optimal policy (for minimum data rate constraints) multiplies the data rate values by the KKT multipliers. These two special cases have been studied in [7]. In general, in the solution to the generalized opportunistic scheduling problem, we affine translate the functions of the data rates and then use an argmax rule to obtain the optimal policy.

An example of a single interface opportunistic scheduling problem with multiple constraints is as follows. Suppose that the system objective of the opportunistic scheduling problem is to maximize the *goodput*. By goodput we mean that we consider the transmission of only user data and not any headers (overhead associated with the transmission). Assume that the header size is of H bytes per transmission. Also assume that all the users specify two constraints. The first constraint is the usual processor sharing constraint with weight ϕ_i for user i, and the second is a minimum goodput rate constraint (*i.e.*, the data rate without headers) of R_i. Thus the problem can be formulated as follows.

$$max_Q \sum_{i\in\mathcal{N}} E\{(\mu_i - H)I_{Q(\vec{\mu})=i}\} \quad such\ that \quad \forall i,\ E\{I_{Q(\vec{\mu})=i}\} \geq \phi_i;\ E\{(\mu_i - H)I_{Q(\vec{\mu})=i}\} \geq R_i$$

The solution of this problem is,

$$Q^* = argmax_i\{\mu_i + \lambda_i^1 + \lambda_i^2(\mu_i - H)\} \quad i.e., \quad Q^* = argmax_i\{\mu_i(1+\lambda_i^2) + \lambda_i^1 - H\lambda_i^2\}$$

The λ's are the KKT multipliers as defined in Theorem 1.

We now discuss various issues related to the calculation of the optimal solution for the opportunistic scheduling problem formulated in Problem P1. The first issue is how to calculate the KKT multipliers in a real system so that we obtain the desired optimal scheduling policy. In [7] the authors present a stochastic approximation algorithm to calculate such constants. The algorithm is outlined in Section IV-D. The basic idea behind this type of algorithms is to start with all zero KKT constants. Then in each timeslot, along a sample path, a correction term proportional to the error term (the difference between the actual constraint bound G_i^j and the so far achieved average constraint value) is

added. A similar approach can be taken to calculate the $\lambda's$ in Equation 1. Such algorithms are not computationally complex and hence can be used in each timeslot. The results of [7] show that these algorithms converge quite fast.

Note that if the system is unfeasible, *i.e.*, the constraints cannot be satisfied then the stochastic approximation method will not converge. Thus we must guess the feasibility heuristically if the stochastic approximation algorithm seems to diverge. In fact, in this sense the optimal solution is limited because it does not give any information about the system feasibility. Also, after the arrival of a new user or after the departure of a user, all the KKT multipliers change and must be recalculated.

The solution for the opportunistic scheduling assumes that all the users are greedy and hence the solution methodology is good for the heavy traffic scenario. Also note that the optimal solution with the processor sharing constraints does not guarantee the *exact* generalized processor sharing (which guarantees the distribution of excess server capacity among the active users in proportion to their weights) if some users are inactive. Also other types of guarantees, *e.g.*, short term fairness constraints and buffer overflow constraints cannot be provided with this approach.

B. Opportunistic Scheduling over Multiple Interfaces

The next generation of wireless communication devices will be equipped with more than one interface. Each interface provides different characteristics. Such devices have been already discussed in the literature but within a context of connectivity and mobility. A device may have more than one interface because a single technology might not be sufficient in terms of coverage or because in different environments different technologies might be more suitable. We consider a general opportunistic scheduling problem over multiple interfaces with long term user QoS constraints (similar to the QoS constraints described in the previous subsection) and physical constraints. The physical constraints are imposed by the system structure and must be satisfied in each timeslot. For example, consider the different interfaces as different physical antennas. There are K physical interfaces (antennas) for each user. The communication bandwidth is divided in K bands and each antenna can be tuned to any band. Thus each antenna can be used for any interface, but an antenna can be used for only one interface at a given time. Thus in the above case, a physical constraint can be specified as follows. If an interface k is assigned to user i in a given timeslot, then another interface l cannot be assigned to user i and another user j cannot be assigned to the interface k in the same timeslot. This is precisely the scenario in the multi-beam satellite systems. In such systems, there are K beams and N earth based stations. A

beam can be used to serve any one of the stations in a given timeslot. At most one beam could be used to serve a station. A more complex example of physical constraint can be a modified version of the above physical constraint as follows. Suppose that the base station is limited by the maximum total power it can use for transmission in any given timeslot. In this example, the base station has to choose a power distribution among the different interfaces such that the total transmitted power must be less than the maximum allowed power consumption. problem in multi-beam satellite networks is considered in [11].)

We denote by $\mathcal{K}$ the set of interfaces and index the interfaces by the letter k. Thus $\vec{\mu}_i = (\mu_i^1, \cdots, \mu_i^K)$ and $[\mu]$ is the $N \times K$ matrix where μ_i^k denotes the current data rate for user $i \in \mathcal{N}$ over interface $k \in \mathcal{K}$. For notational convenience only, we assume that each user has only one QoS constraint (referenced by g_i). A stationary policy $\vec{Q} = (Q^1, \cdots, Q^K)$ denotes a vector function on $[\mu]$, which assigns each interface to a particular user in each timeslot according to the specified rule Q. Though other physical constraints can also be taken into account in a similar fashion, in the following discussion we assume the following physical constraint: any feasible policy cannot assign two users to the same interface or two interfaces to the same user in any timeslot, *i.e.*, $k \neq \hat{k} \Leftrightarrow Q^k([\mu]) \neq Q^{\hat{k}}([\mu])$.

Problem Formulation and the Solution: The multiple interface opportunistic scheduling problem can be defined as follows.

P 2:

$$max_Q \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} E\{f_i(\mu_i^k) I_{Q^k([\mu])=i}\}$$
$$\text{such that } \forall i, \quad \sum_{k\in\mathcal{K}} E\{g_i(\mu_i^k) I_{Q^k([\mu])=i}\} \geq G_i \quad \text{and} \quad k \neq \hat{k} \Leftrightarrow Q^k([\mu]) \neq Q^{\hat{k}}([\mu])$$

In words, the objective of this problem is to maximize the expected system utility subject to the physical constraints and to that on the long term, the user QoS constraints will be satisfied.

Now define the function $Kargmax_{i,k}(f_{i,k})$ which is a solution of the following optimization problem.

$$max_{a_{i,k}} \sum_{i\in\mathcal{N}} \sum_{k\in\mathcal{K}} E\{a_{i,k} f_{i,k}\} \tag{2}$$
$$s.t.\ a_{i,k} \in \{0,1\}, \quad k \neq \hat{k} \Leftrightarrow a_{i,k} + a_{i,\hat{k}} \leq 1 \quad \text{and} \quad i \neq \hat{i} \Leftrightarrow a_{i,k} + a_{\hat{i},k} \leq 1$$

In words, $Kargmax_{i,k}(f_{ik})$ is a function defined as follows: Choose at the most K entries from the matrix $[f_{ik}]$ of $N \times K$ entries, satisfying the physical constraints, such that the sum of the selected terms is maximum. For the constraints we have considered, we choose a maximum of one entry per column and per row. Hence, this special case

of $Kargmax_{i,k}$ is an assignment problem, or a weighted bipartite graph matching problem. The nodes from one part of the graph represent users and the nodes from another part represent the interfaces. The weights of each link are specified by the term $f_{i,k}$.

Theorem 2: The solution $\vec{Q}^*$ of the multiple interface constrained opportunistic scheduling problem defined in Problem P2, if one exists, is of the following form.

$$\exists \lambda_i \geq 0 \;\; s.t. \;\; \vec{Q}^* = Kargmax_{i,k}\{f_i(\mu_i^k) + \lambda_i g_i(\mu_i^k)\} \quad \text{where} \quad \sum_{k\in\mathcal{K}} E\{g_i(\mu_i^k) I_{Q^k([\mu])=i}\} > G_i \Rightarrow \lambda_i = 0 \tag{3}$$

Where, the $Kargmax_{i,k}(f_{ik})$ is the function defined as above.

Proof: Consider any feasible policy Q. Then there exist nonnegative constants λ_i such that the following holds.

$$\begin{aligned}
\sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} E\{f_i(\mu_i^k) I_{Q^k=i}\} &\leq \sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} E\{f_i(\mu_i^k) I_{Q^k=i}\} + \sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} \lambda_i \Big[E\{g_i(\mu_i^k) I_{Q^k=i}\} - G_i\Big] \\
&= \sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} E\Big[\big(f_i(\mu_i^k) + \lambda_i g_i(\mu_i^k)\big) I_{Q^k=i}\Big] - \sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} \lambda_i G_i \\
&\leq \sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} E\Big[\big(f_i(\mu_i^k) + \lambda_i g_i(\mu_i^k)\big) I_{Q^{*k}=i}\Big] - \sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} \lambda_i G_i \qquad (4) \\
&= \sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} E\{f_i(\mu_i^k) I_{Q^{*k}=i}\} + \sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} \lambda_i \Big[E\{g_i(\mu_i^k) I_{Q_{*k}=i}\} - G_i\Big] \\
&= \sum_{\substack{i\in\mathcal{N}\\k\in\mathcal{K}}} E\{f_i(\mu_i^k) I_{Q^{*k}=i}\}
\end{aligned}$$

(4) follows from the definition of $\vec{Q}^*$ in (3) and by the property of the $Kargmax$ function defined in (2). ■

Thus the same solution structure is carried over from the optimal policy for the single interface opportunistic scheduling problem to the optimal policy for the multiple interface opportunistic scheduling problem. The constants $\lambda's$ could be calculated with the same stochastic approximation algorithm as the one discussed in the previous section on the single interface problem. We note that even when the constants in the optimal solution are given, $Kargmax$ solves an assignment problem in every timeslot. Even for simple constraints the optimal policy for the multiple interface opportunistic scheduling problem involves complex computations in each timeslot. As opposed to this, the optimal policy for the single interface opportunistic scheduling problem simply selects the largest entry. In a special case where each user has the same data rate over all the interfaces in every timeslot (different users still have different data rates) the optimal policy is of the form $Kargmax_i(f_i(\mu_i) + \lambda_i g_i(\mu_i))$, where $Kargmax$ becomes a function which simply

selects the K highest entries. We note that for different physical constraints, $Kargmax$ is a different function and the solution of an optimization problem in itself. For example, for the physical constraints such as the power allocation constraints discussed earlier, $Kargmax$ is a function which chooses a power allocation strategy in each timeslot such that the sum of the selected $f_i(\mu_i^k) + \lambda_i g_i(\mu_i^k)$ terms is maximum. In such cases, we could either use to solve this problem an exact but complex optimization approach or a simpler but approximate approach.

C. Throughput Optimal Scheduling Rule over Multiple Interfaces

In this subsection only, we do *not* assume that the users are greedy. Hence the base station might not always have data to be sent on the downlink to a given user. We assume that the data to be transmitted to the users is queued at the base station on separate queues (one queue per user). In this section, when we discuss stability we mean that all the queues should remain finite. It is important to design scheduling policies that yield stability. It may not be always possible (because of the traffic generated for some users could be too large) but if there exists a scheduling policy that yields stability, we want to find it. Intuitively this can be ensured if, for each queue, the service rate is greater than the incoming traffic rate. In simple static queuing systems this translates into ensuring that each user gets a time share of the server large enough to yield a service rate greater than its incoming traffic rate. But in time-varying dynamic systems, ensuring only this is not enough as there is no simple linear relation between the amount of the processor share and the corresponding service rate a user gets. Hence, it is also important to *optimally time* the service offering for a user when the channel conditions are good for that user. Finding such stable policies might depend on knowing the average incoming traffic rates for each user and any method which requires the exact incoming rates to be known to obtain a stable scheduling policy critically depends on estimating the incoming rates. Throughput optimal scheduling policies on the other hand, are policies which guarantee stability under any incoming traffic rates, as long as there is at least one such stable scheduling policy. Thus for these policies there is no need to estimate any incoming rate. Note that in this kind of problems, there are no per user QoS constraints to satisfy.

For the single interface case a throughput optimal scheduling rules has been given in [1]. In this section we generalize the throughput optimal scheduling rules termed as M-LWWF, *i.e.*, Modified Largest Weighted Work First to multiple interface systems. The rule for the single interface case is stated as follows: In any given timeslot, serve a user having the maximum $a_i q_i(t)\mu_i(t)$ with the a_i's are any positive (nonzero) constants and $q_i(t)$ is the queue size for a user i at

time t. Thus $Q^* = argmax_i\{a_i q_i \mu_i\}$ is a throughput optimal scheduling rule for the single interface case.

For the multiple interfaces case we generalize this result as follows.

Theorem 3:

$$\vec{Q^*}(t) = Kargmax_{i,k}\{a_i q_i(t) \mu_i^k(t)\} \tag{5}$$

is a throughput optimal scheduling rule for the multiple interface case, *i.e.*, if there exists any scheduling rule that makes all user queues stable then the above rule also makes all user queues stable.

Proof: Refer to the appendix in [5]. The details are omitted due to lack of space. ■

Note that we do not assume that the users are greedy as we have assumed in Theorems 1 and 2. Thus this throughput optimal scheduling rule, on a long term, provides a service rate at least equal to the incoming rate for each user irrespective of the channel conditions assuming that it is at all possible. Note that this throughput optimal scheduling policy is different from the optimal opportunistic scheduling policy as given in equation (3) with QoS constraints based on the average data rates. For throughput optimal scheduling rules, if one user misbehaves it affects all the other users performance, possibly making all the queues unstable. But the policy given in equation (3) is robust against misbehavior of some users. In such cases it still provides the required data rate for the remaining users. For more details on throughput optimal rules refer to [2, 11, 14].

D. Numerical Results

We simulated the opportunistic scheduling policies for typical settings of a CDMA-HDR (1xEV-DO) system. The data rate for each user is determined by the Signal to Noise Ratio (SNR) as shown in Table I taken from [3].

The SNR for each user is modeled as an autoregressive log-normally distributed channel. Specifically, $s_i(t+1) = \gamma s_i(t) + (1-\gamma) n_i(t+1)$; where $s_i(t)$ denotes the channel SNR (db) in timeslot t for user i and the $n_i(t)$ denote the channel variations (noise terms) assumed to be normally distributed independent rv's. These $n_i(t)$ rv's have a standard deviation of 15 in all cases. γ, the auto-regression coefficient is set to 0.7.

We use a stochastic approximation method to calculate the values of the constants λ's. For simplicity assume that each user has only one constraint (g_i). Then a simple stochastic approximation algorithm to calculate λ_i is given as follows: We start with $\lambda_i(0) = 0$ for all i. After each timeslot t we modify the $\lambda_i(t)$ values as follows: $\lambda_i(t+1) =$

TABLE I

DATA RATE VS SNR FOR AN HDR USER

SNR (db)	-12.5	-9.5	-8.5	-6.5	-5.7	-4.0
Data rate (kbps)	38.4	76.8	102.6	153.6	204.8	307.2

SNR (db)	-1.0	1.3	3.0	7.2	9.5
Data rate (kbps)	614.4	921.6	1228.8	1843.2	2457.6

TABLE II

SIMULATION DETAILS FOR CASE 1. SYSTEM THROUGHPUT=1258 KBPS

User	Mean SNR (db)	Variance SNR (db)	Requested ϕ_i	Requested Rate (kbps)	Served ϕ_i	Served Rate (kbps)	λ_{ϕ_i} λ_{ϕ_i}	λ_{R_i} λ_{R_i}
0	5	15	0.3	700	0.401	813	0	0
1	-2	15	0.6	200	0.599	445	971	0

$\lambda_i(t) + \big(g_i(\mu_i) I_{Q^*(t)=i} - G_i\big)\delta$, where δ is a small step-size. A more detailed discussion on the implementation of stochastic algorithms for opportunistic scheduling policies can be found in [7].

We present simulation results for three cases: two related to our opportunistic scheduling problem over a single interface with multiple constraints, and one related to our scheduling problem over multiple interfaces. In the first two cases, each user has two constraints. The first constraint is a processor sharing constraint and the second constraint is a data rate constraint. In the third case each user has only one processor sharing constraint.

Case 1: There are two users in the system and the results of the simulation are presented in Table II. The simulation is run for $100,000$ timeslots. The simulation results show various interesting aspects of the opportunistic scheduling problem. First note that user 0 has a much better channel than user 1. The processor share requirement of user 0 is much less than that of user 1 while its rate requirement is much larger than that of the user 1. The QoS requirements of both the users are satisfied which means that the system is feasible. The processor share given to user 0 is much larger than its requirement and hence the λ associated with the processor share constraint for user 0 is zero as expected. Similarly the λ associated with the rate constraint for user 0 is zero. For user 1 the processor share given is just equal to the required processor share. Hence the λ associated with it is nonzero. But as the rate constraint associated with user 1 is satisfied by a large margin the λ associated with it is zero.

Case 2: Now we simulate a system with 10 users to study the effect of increased number of users of the performance of optimal opportunistic scheduling policy. The user details are given in Table III.

In this simulation all the even numbered users have identical requirements (a processor share requirement of 0.06 and a data rate requirement of 70 $kbps$) and they experience a better channel than the other users. All the odd numbered users have identical requirements (a processor share requirement of 0.12 and a data rate requirement of 20 $kbps$).

We note that the results of the simulation are similar to those of the previous simulation. As there are ten users

TABLE III

SIMULATION DETAILS FOR CASE 2. SYSTEM THROUGHPUT=1972 KBPS

User	Mean SNR (db)	Variance SNR (db)	ϕ_i Requested	Rate (kbps) Requested	ϕ_i Served	Rate (kbps) Served	λ_{ϕ_i} λ_{ϕ_i}	λ_{R_i} λ_{R_i}
0	5	15	0.06	70	0.072	174	0	0
1	-2	15	0.12	20	0.119	198	1222	0
2	5	15	0.06	70	0.073	176	0	0
3	-2	15	0.12	20	0.119	198	1186	0
4	5	15	0.06	70	0.073	176	0	0
5	-2	15	0.12	20	0.119	196	1241	0
6	5	15	0.06	70	0.113	276	0	0
7	-2	15	0.12	20	0.119	196	1217	0
8	5	15	0.06	70	0.075	181	0	0
9	-2	15	0.12	20	0.119	196	1244	0

TABLE IV

SIMULATION DETAILS FOR CASE 3.

User	Mean SNR Interface 0,1 (db)	Mean SNR Interface 2,3 (db)	ϕ_i Requested	ϕ_i Served	λ_{ϕ_i}	Rate Served (kbps)
0	-2	-2	0.5	0.48	426	749
1	-2	-2	0.5	0.48	458	753
2	-2	2	0.5	0.49	6	933
3	-2	2	0.5	0.49	151	940
4	2	-2	0.5	0.49	45	934
5	2	-2	0.5	0.49	49	940
6	2	2	0.5	0.53	0	1081
7	2	2	0.5	0.54	0	1115

in the system, the system throughput has been increased due to the increased multi-user diversity gain. Also the λ's associated with both the constraints for all the even numbered users are zero as both the constraints are satisfied by a large margin. But the λ's associated with the processor sharing constraint for the odd numbered users have nonzero (large) values as expected. Note that the even numbered users have not all received the same extra processor share. This shows that the opportunistic scheduling policy has not distributed the remaining extra processor share evenly among all the users (which would be the case for GPS).

Case 3: Now we simulate the opportunistic scheduling policy (solution to problem P2) for a multiple interface system with four interfaces and eight users. The simulation parameters and the results are given in Table IV. While the users $0, 1$ have a bad channel (mean SNR = -2 db) for all the interfaces, the users $6, 7$ have a good (mean SNR = 2 db) channel on all the interfaces. The users $2, 3, 4, 5$ have a bad channel for two interfaces and a good channel for the other two interfaces. As before, for all users the SNR variance is fixed to 15 while the auto-regression coefficient is set to 0.7. All users require a processor sharing constraint with $\phi_i = 0.5$, *i.e.*, on an average each user must be selected on any interface once in two timeslots. As expected using our policy all of the constraints are approximately satisfied. Because the system is just feasible (*i.e.*, $\sum \phi_i = 4$) there are slight discrepancies between the ϕ_i requested by the users and ϕ_i served by the policy due to the stochastic behavior of channel SNR. Note that the λ's associated with users $6, 7$ are zero while those associated with users $0, 1$ are large to balance out the effect of different channel conditions for different users. The λ's for users $2, 3, 4, 5$ are small compared to those for users $0, 1$ and most of the time they get

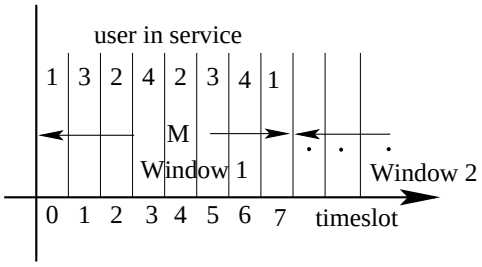

Fig. 3. A scheduling policy conforming to short term fairness constraints.

user	0	1	2	3	4	5	6	7	8	9
ϕ_i	0.05	0.15	0.05	0.15	0.05	0.15	0.05	01.5	0.05	0.15
mean SNR (db)	-4.0	-4.0	-2.0	-2.0	0.0	0.0	2.0	2.0	4.0	4.0

TABLE V

USER DETAILS FOR THE SECOND SIMULATION SCENARIO

served on the interfaces on which they have a better channel.

IV. OPPORTUNISTIC SCHEDULING WITH SHORT TERM FAIRNESS CONSTRAINTS

In this section we only consider processor sharing constraints, though our approach can be generalized for other types of QoS constraints. The users are assumed to be greedy.

A. Problem Formulation

Let us assume that each user i has an associated weight ϕ_i such that $\sum_{i\in\mathcal{N}} \phi_i \leq 1$. Let us group the timeslots into *successive non-overlapping* windows of M timeslots each. Then a scheduling policy in which every user gets service for at least $M\phi_i$ timeslots in any such window is said to follow the *short term fairness constraint* with STF (short term fairness) window of size M (refer to Figure 3). Note that Weighted Round Robin which allocates $M\phi_i$ consecutive timeslots to user i is an example of such a policy. The maximum number of timeslots between two consecutive service offerings for a user is called the *starvation period* for that user. The objective of our opportunistic scheduling problem is to maximize the system throughput. Thus the opportunistic scheduling problem with short term fairness constraints (with a STF window of size M) can be formulated as the following optimization problem.

P 3: $$max \sum_{t=0,i\in\mathcal{N}}^{M-1} E[\mu_i(t) I_{Q(t)=i}] \qquad \text{such that } \forall i,\ \sum_{t=0}^{M-1} I_{Q(t)=i} \geq M\phi_i$$

Henceforth we shall assume that $\sum_{i\in\mathcal{N}} \phi_i = 1$ though the case $\sum_{i\in\mathcal{N}} \phi_i < 1$ can be handled in a similar manner. Also we assume that $M\phi_i$ is an integer for all i. This limits the values of M and the possible values of ϕ_i's. In words, the above optimization problem can be stated as follows: Among all scheduling policies which select each user $M\phi_i$ times in M consecutive timeslots find the one which maximizes the system throughput. A policy which satisfies the above

short term fairness constraints (with a STF window of size M) guarantees that no user will experience a starvation period greater than $2M - 1$ and each user will get its fair share of $M\phi_i$ timeslots in successive non-overlapping windows of M timeslots.

We are not imposing any structure on the channel characteristics. The optimal scheduling policy will depend on many parameters including the channel state model, the current channel state for each user, the number of users, the STF window size M, the current timeslot, and the state of the short term fairness constraints. In real systems it is difficult to know (or estimate) many of the parameters involved in the channel state model. Hence in general the solution of the above opportunistic scheduling problem is difficult to obtain. However we can simplify the above problem into some special cases and study the optimal scheduling policy for these special cases. We do that in the next section.

B. Special Cases

1) $\underline{M = \infty}$*:* Let us consider the special case in which the STF window (M) is equal to ∞. This special case can be thought of as a long term processor sharing fairness constraint which is defined as follows: On an average, user i must get a time share of the base station that is greater than or equal to ϕ_i. This obviously does not guarantee anything over a finite time period. Then the opportunistic scheduling problem with this long term fairness constraint can be defined as follows:

$$max \sum_{i \in \mathcal{N}} E\{\mu_i(t) I_{Q(\vec{\mu}(t))=i}\} \qquad \text{such that } \forall i,\ E\{I_{Q(\vec{\mu})=i}\} \geq \phi_i$$

It has been shown in Section III-A that the above problem has a solution of the form $Q^*(t) = argmax_i\{\mu_i(t) + \lambda_i\}$ (with $\lambda_i > 0 \Rightarrow E\{I_{Q(\vec{\mu})=i}\} > \phi_i$). λ_i is the non-negative Lagrange multiplier associated with the constraint of user i. We call this policy the Long-Term (LT) optimal scheduling policy. We have described a method to estimate the constraints λ_i's in a real system in Section III-D.

2) $\underline{N \text{ i.i.d. users: } \forall i, \phi_i = 1/N,\ M = N}$*:* This case considers the shortest possible STF window. When all users have the same weight ϕ_i, then the smallest STF window size is equal to the number of users N. We assume that the data rates for all the users are identically and independently distributed (i.i.d.) in each timeslot. We also assume that

$M = N$ and $\forall i, \phi_i = 1/N$. Thus our original optimization problem P3 becomes,

$$max \sum_{t=0,}^{N-1} \sum_{i \in \mathcal{N}} E[\mu_i(t) I_{Q(t)=i}] \quad s.t. \forall i, \ \sum_{t=0}^{N-1} I_{Q(t)=i} = 1 \tag{6}$$

In words the above optimization problem can be stated as follows: Find a scheduling policy which selects each user once in N consecutive timeslots and maximizes the system throughput. Let Q^* be an optimal policy for the above optimization problem when the data rates for all the users are i.i.d. in each timeslot. Also define, $A^*(0) = \mathcal{N}$ and $A^*(t) = A^*(t-1) - Q^*(t-1)$ where $A^*(t)$ denotes the set of unserved users at time t for the current window, and $\mathcal{A}^*(t) - Q^*(t)$ denotes the relative complement of $Q^*(t)$ w.r.t. $\mathcal{A}^*(t)$. Then we claim,

Theorem 4: $\forall t = \{0, \cdots, N-1\}, \ \ Q^*(t) = argmax_{i \in A^*(t)}\{\mu_i(t)\}$ is the optimal opportunistic scheduling policy for the problem P3 with $M = N, \ \phi_i = 1/N$ and i.i.d. users. We call this policy the Opportunistic Round Robin policy.

In words the optimal opportunistic scheduling policy selects at the beginning of a new STF window, the user with the highest data rate. Then it considers this user to be inactive till all the other users get selected (*i.e.*, in this special case till the end of the window). Once all the users get selected once in a window of M timeslots, the policy considers all the users to be active once again and repeats this whole procedure.

Proof: Let $Q = (Q(0), \cdots, Q(N-1))$ be any other feasible policy. Then, $\mu_{Q^*(0)}(0) \geq \mu_{Q(0)}(0)$ by the choice of Q^*. Now $Q^*(1)$ operates on $A^*(1) = \mathcal{N} - Q^*(0)$ and $Q(1)$ operates on $A(1) = \mathcal{N} - Q(0)$. But, $A(1)$ and $A^*(1)$ have the same number of unserved users and hence are statistically similar due to the assumption of identical users (to satisfy the fairness constraint the served user cannot be served again before the end of the window.) Thus $E(\mu_{Q^*(1)}) \geq E(\mu_{Q(1)})$, and so on $\forall t$. ■

3) N independent users: $\forall i, \phi_i = 1/N, \ M = N$: In this case we relax the assumption that all the users are identical. We assume that the user data rates are independent of each other and also across time. Under these conditions, we claim that there exist $2^N - 2$ constants and an associated $argmax$ decision policy which is an optimal opportunistic scheduling policy for the problem P3.

Theorem 5: If $\phi_i = 1/N$, and if the data rates of all the users are independent of each other and across time then

$$Q^*(t) = argmax_{i \in A^*(t)}\{\mu_i(t) + V^*_{\mathcal{A}^*(t)-\{i\}}\} \tag{7}$$

$$A^*(0) = \mathcal{N}, A^*(t+1) = A^*(t) - Q^*(t)$$

is the optimal opportunistic scheduling policy for the optimization problem P3. Here, $V^*_{\mathcal{B}}$ is the optimum expected reward value, *i.e.*, the sum of the data rates of the selected users in all of the timeslots in a window of $|\mathcal{B}|$ timeslots, associated with the optimal policy for a problem P3, with a set of $\mathcal{B}$ users, and with $\phi_i = 1/|\mathcal{B}|, M = |\mathcal{B}|$.

In words the above optimal policy can be described as follows. Suppose at the beginning of the timeslot t the set of unserved users is $A^*(t)$. Now suppose a policy selects user $i \in A^*(t)$ for service in this timeslot. Then the system would get a reward (*i.e.*, a data rate) of $\mu_i(t)$ in this timeslot. In the remaining window of $|A^*(t)| - 1$ timeslots the expected optimal reward that the system would get is (by definition) $V^*_{\mathcal{A}^*(t)-\{i\}}$. Hence the optimal policy in this timeslot selects a user with the maximum $\mu_i(t) + V^*_{\mathcal{A}^*(t)-\{i\}}$ value to maximize the total expected data rate. Note that the constants $V^*_{\mathcal{B}}$ can be thought as the expected total reward, *i.e.*, the sum of the data rates in a window of $|\mathcal{B}|$ timeslots that the system would get by following an optimal policy. Also note that these constants are not similar to the constants λ_i from the first special case in IV-B.1. Only one λ_i constant is associated with a user i in the optimal long term policy, while $V_{\mathcal{A}^*(t)-\{i\}}$ depends on the user i, the time index, and $\mathcal{A}^*(t)$. There are more than one such constants associated with each user. We prove our claim by constructing such a policy recursively.

Proof: Clearly if $\mathcal{N} = \{i\}$, *i.e.*, for a singleton set the policy $Q^*(t) = i$, *i.e.*, selecting user i in every timeslot is trivially optimal. Hence we assign the optimal reward associated with this single user system $V^*_i = E(\mu_i)$. Now consider a two user system with $M = 2, \phi_0 = \phi_1 = 1/2$. In timeslot 0 if the policy selects user 0 then the total expected reward in the current STF window would be $\mu_0(0) + E(\mu_1(1))$ otherwise $\mu_1(0) + E(\mu_0(1))$. Clearly the optimal policy should select user 0 if $\mu_0(0) + E(\mu_1(1)) > \mu_1(0) + E(\mu_0(1))$ or otherwise select user 1. Similarly, the rest of the proof follows from an induction argument. Suppose that for the set of users $\mathcal{N}$ the policy Q^* as defined in Equation 7 is optimal. If we add another user then the optimal policy would be to choose a user in timeslot 0 optimally and then use the optimal policy for the remaining N users in the next N timeslots. This claim is valid because we assume the independence across the users and time. The theorem follows directly after this claim and the recursive definition of set $\mathcal{A}^*(t)$. ■

For specifying the optimal policy we need $2^N - 2$ constants, *i.e.*, the number of (unordered) subsets of $\mathcal{N}$ minus 2 (corresponding to the null set and the set $\mathcal{N}$). Thus this optimal policy is much more complex than the optimal policy in the previous case (under the identical users assumption). Theoretically it is possible to calculate the $V_{\mathcal{A}^*(t)-\{i\}}$

constants given the distribution of the data rate for each user. Estimates of theses constants which asymptotically converge to the true constants can be obtained along a sample path.

C. Heuristic Policy

Until now we have analyzed three special cases of the opportunistic scheduling problem P3. The long term optimal policy selects a user having the maximum "$\mu_i(t) + \lambda_i(t)$" in each timeslot while the first (respectively the second) special short-term optimal policy selects a user with the maximum "$\mu_i(t)$" (resp. "$\mu_i(t) + V^*_{\mathcal{A}^*(t)-\{i\}}$"). The long term policy adds a bias to the data rate values while the short-term policies remove a user from the set of active users if it has got its fair share in the current STF window. This motivates us to define the following heuristic policy for the general opportunistic scheduling problem P3. The Heuristic Policy (HP) with a STF window of size M is defined as follows.

$$\forall t = \{0, \cdots, M-1\}, \;\; Q(t) = argmax_{i \in \mathcal{A}(t)}\{\mu_i + \lambda_i\} \tag{8}$$

where the λ_i's are the constants derived from the LT policy. $A(0) = \mathcal{N}, \;\; N_i(0) = 0$, and $N_i(t), A(t)$ are defined recursively as $N_i(t) = N_i(t-1) + I_{Q(t-1)=i}, \;\; A(t) = A(t-1) - Q(t-1)I_{N_i(t)=M\phi_i}$.

The Heuristic Policy can be described with the following steps.

- *Step 1*: Initialization at the beginning of a new STF window: The set of initial *active* users is the set $\mathcal{A}(0) = \mathcal{N}$. The fair share of user i is initialized to $M\phi_i$.
- *Step 2*: User selection: In each timeslot the user from the set of *active* users $A(t)$, having the largest $\mu_i(t) + \lambda_i$ value is selected for service.
- *Step 3*: Book-keeping: A counter that keeps track of how much service (*i.e.*, number of timeslots) the selected user has got in the current window is incremented by one. If the counter is equal to the fair share of that user then that user is removed from the set of active users. Step 2 is then repeated for the next timeslot.

At the end of the current STF window the Heuristic Policy restarts from Step 1 with a new non-overlapping STF window.

D. Numerical Results

Now we compare the Long Term policy (LT) and our Heuristic Policy (HP) in terms of average system throughput and short term fairness. We simulate these policies for HDR users.

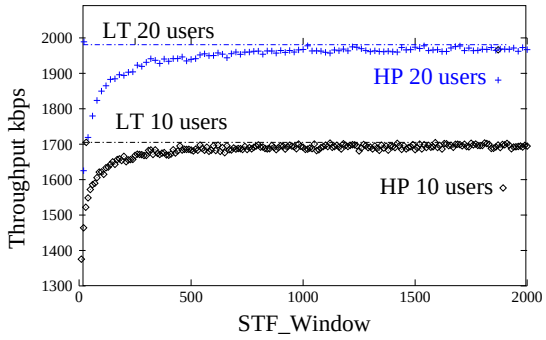

Fig. 4. Throughput vs STF window for the LT and the HP policies for i.i.d. HDR users, no. of timeslots = 10,000.

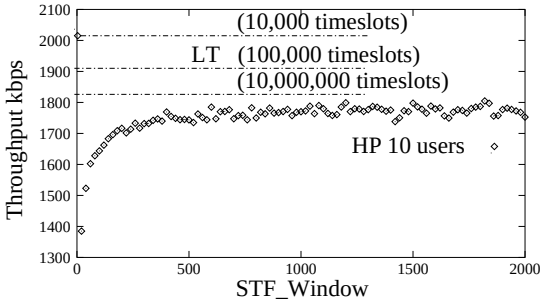

Fig. 5. Throughput vs STF window for the LT and HP policies. 10 HDR users with different channel conditions and different ϕ_i's, no. of timeslots = 10,000.

The implementation of HP simulates the long term optimal policy to calculate the constants λ_i's on the fly which is not computationally heavy. We have simulated a heuristic similar to HP but with $\lambda_i = 0$ for all i to understand the importance of the λ_i's. We observed that the throughput of this heuristic is much lower than the throughput of the HP which shows the importance of the λ_i's.

We first consider the case of i.i.d. HDR users with equal ϕ_i's. We assume that the SNR values for each user in each timeslot are independent and identical log-normal random variables (rv) with mean 0 and standard deviation 5 (for more details please refer to [3, 7]). The data rate for each user is determined by the corresponding SNR value according to Table I. The throughput versus STF window (M) curves obtained via simulation for 10 (and 20) users with equal ϕ_i's over 10,000 timeslots are shown in Figure 4. (Note that we also simulated the Weighted (non opportunistic) Round Robin policy and found that it has a very low throughput (720 kbps in this scenario) compared to the other opportunistic policies.)

We make the following observations. The throughput of HP increases as the short-term fairness window increases as expected. After a particular value of M, increasing the STF window does not increase the throughput by a large value, *i.e.*, the throughput reaches the saturation stage. But increasing the window size beyond this value only increases the maximum guaranteed starvation period which is equal to $2M - 1$. Hence ideally the STF window should not be greater than this knee value. As the number of the users in the system increases the average system throughput also increases due to the multi-user diversity (but the throughput per user decreases).

To understand the behavior of these policies under realistic channel conditions, we consider a case when the users

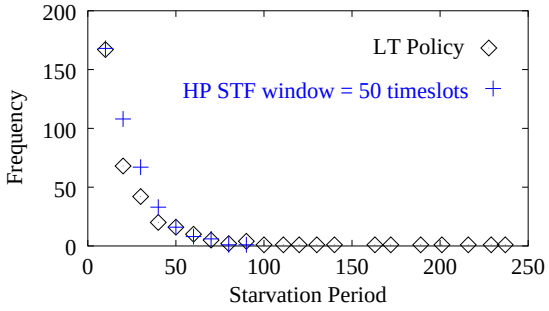

Fig. 6. Frequency distribution of the starvation period for the 10 HDR users with different channel conditions and different ϕ_i's, no. of timeslots = 10,000.

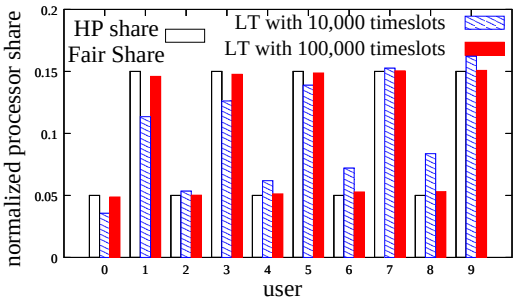

Fig. 7. Normalized processor share given to each user by the LT and the HP policies. 10 HDR users with different channel conditions and different ϕ_i's

have different channel distributions and different ϕ_i's. We assume that there are 10 users (see Table V) in the system. The channel SNR for each user is modeled as an autoregressive log-normally distributed channel as in section III-D. Users $0, 1$ have mean SNR of -4, users $2, 3$ of -2, and so on. γ, the auto-regression coefficient is set to 0.7. The users with the same mean SNR have different ϕ_i of either 0.05 or 0.15. Thus, user 0 has a $\phi_0 = 0.05$ and user 1 has a $\phi_1 = 0.15$. The throughput versus M is plotted in Figure 5. Figure 6 plots the empirical frequency distribution of the starvation periods. For each policy we calculate the number of times a particular value of the starvation period is experienced by the users. Thus the y axis value of 70 versus the x axis value of 30 would mean that all users cumulatively have experienced a starvation period of 30, 70 times in the course of the simulation. (We have plotted this graph for starvation period values up to 250 only to show the details clearly. However the tail for the LT policy goes up to 400.) From Figure 6 we notice that LT policy does not give any guarantees on the maximum starvation period. HP with $M = 50$ does guarantee a starvation period of less than 99 timeslots while with LT, users would experience starvation periods of up to 400 timeslots or even more (up to 1000) if the channels are highly correlated ($\gamma = 0.9$) or if the simulation is run for a longer duration, or if there are more users.

From Figure 5, we observe that increasing the STF window size increases the throughput of HP. But there is a large difference between the average system throughput of the LT policy (run for 10,000 timeslots) and the average system throughput for HP even with large STF window sizes. This may suggest that HP is not good in terms of maximizing the average system throughput. Hence we look at the processor share each user has received (*i.e.*, the number of timeslots each user has received) under the LT policy and HP (see Figure 7). We notice that LT policy is biased towards users

with relatively low ϕ_i and better channel. Notice that the LT policy offers more timeslots to users $4, 6, 8, 9$ than their fair share at the expense of users $0, 1$. The reason behind this unfair behavior of the LT policy is as follows. The simulation starts with $\forall i,\ \lambda_i = 0$. Hence the LT policy selects a user having a better channel and lower ϕ_i more often than its fair share during the initial stages. Because the simulation is run only for 10,000 timeslots (instead of an infinite number of timeslots) this unfair behavior at the initial stage leads to unfair behavior of the long term policy over a finite number of timeslots. This also explains the large difference between the throughput of the HP and the long term optimal policy in Figure 5 even for large values of STF window M. Note that as the duration of the simulation increases the throughput of the LT policy decreases and there is not much difference between the throughput for the LT policy with 10,000,000 timeslots and the HP policy with relatively large M. In real systems we expect that the users have finite activity periods (10,000 timeslots would correspond to 17 seconds in a HDR system and 100,000 to approximately 3 minutes). Hence the long term policy is not necessarily a fair policy on any reasonable finite horizon. We also note that the set of active users can change (rather frequently in real systems) and that the channel conditions may also change (non-stationarity); in such situations the LT policy will be even more unfair.

V. Conclusions

In this paper, we studied several opportunistic scheduling problems. We generalized the opportunistic scheduling problem given in [7] to include multiple general user QoS requirements and showed that the optimal solution is an Index policy. We then studied an opportunistic scheduling problem for multiple interface systems with QoS constraints and the physical constraints imposed by the system structure. We showed that the optimal policy for this problem is also an Index policy. We then generalized the throughput optimal scheduling policy to the case of multiple interface systems with general physical constraints. To guarantee short term fairness we defined an opportunistic scheduling problem and proposed a heuristic opportunistic scheduling policy to solve it. Via simulations we compared its performance to the performance of the long term optimal policy and concluded that the throughput performance of the heuristic policy is comparable to that of the long term optimal policy while guaranteeing short term fairness.

Acknowledgments

This work was supported in part by the Indiana Twenty-First Century Fund through the Indiana Center for Wireless Systems and Applications.

REFERENCES

[1] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijaykumar, and P. Whiting. CDMA data QoS scheduling on the forward link with variable channel conditions. Technical report, Bell Laboratories, Lucent Technologies, 2000.

[2] M. Armony. *Queuing Networks with Interacting Network Resources*. PhD thesis, Stanford University, 1999.

[3] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi. CDMA/HDR: A bandwidth-efficient high-speed wireless data service for nomadic users. *IEEE Communication Magazine*, 38:70–77, July 2000.

[4] M. Gurelli and R. Etkin. Capacity simulation of CDMA2000 1xEV-DO forward link with opportunistic beam forming. In *Proceedings of IEEE GLOBECOM*, December 2003.

[5] S. S. Kulkarni and C. Rosenberg. Opportunistic scheduling for wireless systems with multiple interfaces and multiple constraints. Proceedings of ACM/SIGCOM MSWiM, September 2003.

[6] S. S. Kulkarni and C. Rosenberg. Opportunistic scheduling policies for wireless systems with short term fairness constraints. In Proceedings of IEEE GLOBECOM, December 2003.

[7] X. Liu, E. K. P. Chong, and N. B. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, 41(4):451–474, Oct 2003.

[8] Y. Liu, S. Gruhl, and E. Knightly. WCFQ: An opportunistic wireless scheduler with statistical fairness bounds. *IEEE Transactions on Wireless Communication*, September 2003.

[9] Y. Liu and E. Knightly. Opportunistic fair scheduling over multiple wireless channels. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, USA, 2003.

[10] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7:473–489, August 1999.

[11] M. J. Neely, E. Modiano, and C. E. Rohrs. Power and server allocation in a multi-beam satellite with time varying channels. In *Proceedings of IEEE INFOCOM*, New York, USA, 2002.

[12] T. Ng, I. Stoica, and H. Zhang. Packet fair queuing algorithms for wireless networks with location dependent errors. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, USA, 1998.

[13] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate Ad-Hoc networks. In *Proceedings of ACM MOBICOM*, 2002.

[14] S. Shakkottai and A. Stolyar. Scheduling algorithms for a mixture of real-time and non-real-time data in HDR. In 17^{th} *International Teletraffic Congress*, September 2001.

[15] P. Viswanath, D. N. C. Tse, and V. Anantharam. Asymptotically optimal waterfilling in vector multiple access channels. *IEEE Transactions on Information Theory*, 47, January 2001.

[16] P. Viswanath, D. N. C. Tse, and R. Laroria. Opportunistic beamforming using dumb antennas. *IEEE Transactions on Information Theory*, 48:1277–1294, June 2002.