Opportunistic Scheduling Policies for Wireless Systems with Short Term Fairness Constraints

Sunil Suresh Kulkarni and 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 a scheduling problem for packet based wireless 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 the wireless system performance. Such scheduling mechanisms are called opportunistic. In this paper we formulate an opportunistic scheduling problem with *short term processor sharing fairness constraints* as an optimization problem where short term refers to the time window on which the fairness is guaranteed. In its most general form, this problem cannot be solved analytically. We £rst solve the above optimization problem for three special cases. We consider the scheduling problem with long term fairness constraints; then we consider the scheduling problem for the shortest possible window under two sets of assumptions namely, one in which users have identically distributed channel conditions and another in which users have independent channel conditions. Observing the form of the corresponding optimal policies, we de£ne a heuristic policy for our original opportunistic scheduling problem with short term fairness constraints. We show via simulation that our heuristic policy attains a good trade-off by guaranteeing short term fairness while achieving high average system throughput. We also illustrate that the optimal opportunistic scheduling policy with long term fairness constraint is in fact unfair in practical scenarios.**

I. Introduction

Wireless channels have time varying characteristics. Different wireless users perceive different channel quality at the same time because of user shadowing, path losses due to changing environments, and user mobility. These variations in the channel conditions can 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 Scheduling Mechanisms*. If the service requirements of all the users are ¤exible, such opportunistic scheduling methods can result in increased system throughput. CDMA-HDR (IS-856) is an example of High Data Rate system for which opportunistic scheduling mechanisms can take advantage of time varying channel conditions.

In this paper, we consider scheduling at the base station of a packet based wireless cellular system (*i.e*, the downlink) with £xed transmission power (please see Figure 1). The wireless channel for each user differs depending on the location, the surrounding environment, and mobility. We assume that each user reports its downlink channel condition to the base station in a periodic fashion. Thus at a given time, the base station knows the channel condition and hence the data rate it can offer to each user on the downlink. After £nishing a packet transmission the scheduling mechanism at the base station chooses the user to which it will send the next packet. The base station uses the transmission rate as determined by the latest reported channel condition of the selected user. Conceptually one can imagine the base station as having one queue per user and the scheduling mechanism is responsible to choose the next queue to serve based on the QoS constraints and channel condition.

The problem of exploiting channel state variations to increase the throughput of wireless systems has been in focus in recent years. An information theoretic analysis of the capacity of a time-varying channel is presented in [1], [2].

In [3], [4] the authors present 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. In [5], [6] the authors consider different long term QoS constraints and provide an optimal opportunistic scheduling policy. One way of achieving *soft* short term fairness (without strict guarantees) is presented in [5] and the authors compare it with the round robin and optimal long term policies.

In [7], [8] 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. Hence such approach is not useful in practical systems like HDR where the possible data rate could vary in a discrete set taking values from 38.4 to 2457.6 kbps. Clearly assuming that the wireless channel is either good or bad is too restrictive.

In [9] the authors present techniques for optimizing packet data protocols and other network and coding techniques for CDMA-HDR system.

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 starvation of such users. A good scheduling policy must guarantee fair share of the network resources to each user over some £nite time window. There is a need for a policy which takes into consideration the short term requirements of the users. The approach of considering only long term constraints is not suf£cient in practice because optimal long term opportunistic

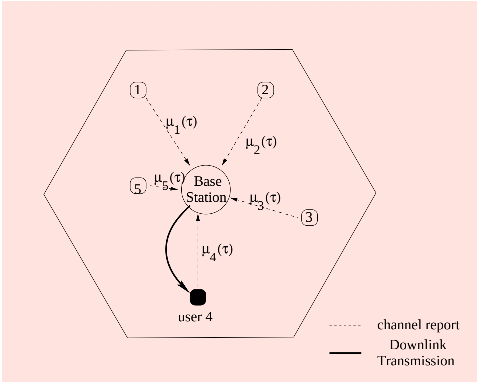

Fig. 1. Opportunistic Scheduling: A typical wireless base station downlink.

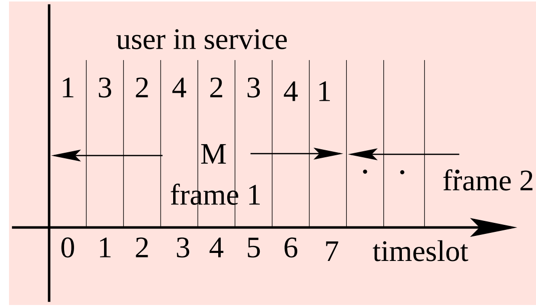

Fig. 2. A scheduling policy conforming to short term fairness consrtaints.

scheduling policies do not give any fairness guarantees in any practical size time windows. We shall illustrate and discuss this point later. Also note that 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. Thus we would like to give a strict guarantee on the maximum starvation period, *i.e.*, the maximum period between two successive service offerings for a active user. Usually this guarantee will be same for all the users. If the minimum possible data rate for a user is strictly greater than zero (or the data rate is constant) then a guarantee on the maximum starvation period would automatically correspond to a minimum data rate for that user.

The paper is organized as follows. In Section II, we de£ne a *strict* 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 III. We provide optimal scheduling policies for these speci£c cases. Getting 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. We then compare our heuristic policy with other policies via simulation in Section V. We conclude in Section VI.

II. OPPORTUNISTIC SCHEDULING PROBLEM WITH SHORT TERM FAIRNESS CONSTRAINTS

We make the following simplifying assumptions to formulate the opportunistic scheduling problem for a time-varying channel with short term fairness constraints. We assume that the system operates on a timeslot by timeslot basis. The timeslot width is £xed and the channel conditions do not vary during a timeslot. 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 as assumed in [1]–[5]. Hence a new scheduling decision has to be taken at the beginning of each timeslot. We also assume that all the transmissions are successful and the base station knows the exact data rate with which it can transmit to each user at the beginning of each timeslot. We assume that all the users are greedy, *i.e.*, each user always has data to receive on the downlink. In this paper we only consider the long term processor sharing constraints as de£ned in [5] and the short term processor sharing constraints, though our approach can be generalized for other types of QoS constraints. We start by introducing the notation.

- $\mathcal{N}$: 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 the users at time t.
- $Q(\vec{\mu}(t))$: This denotes a policy to select a user to serve in timeslot t, given $\vec{\mu}(t)$.

Let us assume that each user has an associated weight ϕ_i (like in Generalized Processor Sharing) and $\sum_{i \in \mathcal{N}} \phi_i \leq 1$. Let us group the timeslots into *successive non-overlapping* frames of M timeslots each (please see Figure 2. Then a scheduling policy in which every user gets service for at least $M\phi_i$ timeslots in any such frame is said to follow the *short term fairness constraint* with STF (short term fairness) window of size M. Note that Weighted Round Robin which serves $M\phi_i$ consecutive timeslots to user i is an example of such a policy. The maximum number of timeslots between two consecutive service offering for a user is called the *starvation period* for that user.

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, 0 otherwise. The objective of the opportunistic scheduling problem is to maximize the system throughput. Hence the opportunistic scheduling problem with short term fairness constraint (with a STF window of size M) can be formulated as the following optimization problem.

$$max \qquad \sum_{t=0, i \in \mathcal{N}}^{M-1} E[\mu_i(t) I_{Q(t)=i}] \tag{1}$$

$$s.t. \forall i, \qquad \sum_{t=0}^{M-1} I_{Q(t)=i} \geq M\phi_i$$

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 users. 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 £nd the one which maximizes the system throughput. A policy which satis£es 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 frames of M timeslots.

We are not imposing any structure on the channel characteristics. The optimal scheduling policy in every timeslot 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 dif£cult 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 dif£cult 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.

III. Special Cases

Let us consider the special case in which the STF window (M) is equal to ∞. This special case can be thought as a long term processor sharing fairness constraint which is de£ned 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 £nite time period. Then the opportunistic scheduling problem with this long term fairness constraint can be de£ned as follows.

$$\begin{aligned} max \qquad & \sum_{i\in\mathcal{N}} E\{\mu_i(t) I_{Q(\vec{\mu}(t))=i}\} \\ s.t.\ \forall i, \qquad & E\{I_{Q(\vec{\mu})=i}\} \geq \phi_i \end{aligned} \tag{2}$$

It has been shown in [5] that the above problem has a solution of the form $Q^*(t) = argmax_i\{\mu_i(t)+\lambda_i\}$ (with $\lambda_i > 0 \Rightarrow E\{I_{Q(\vec{\mu})=i}\} > \phi_i$). The λ_i's are the non-negative Lagrange Multipliers associated with the constraint of the each user. We call this policy the Long-Term (LT) optimal scheduling policy. We describe a method to estimate these constraints λ_i's in a real system in Section V.

The next special case to consider is 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. Thus we consider a special case when $M = N$ and $\forall i, \phi_i = 1/N$. Thus our original optimization problem given in (1) becomes,

$$\begin{aligned} max \qquad & \sum_{t=0,}^{N-1} \sum_{i\in\mathcal{N}} E[\mu_i(t) I_{Q(t)=i}] \\ s.t. \forall i, \qquad & \sum_{t=0}^{N-1} I_{Q(t)=i} = 1 \end{aligned} \tag{3}$$

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. We study this special case under two different assumptions. First we assume that the data rates for all the users are identically and independently distributed (i.i.d.) in each timeslot. De£ne, $A^*(t)$ as the set of unserved users at time t.

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

where, $\mathcal{A}^*(t) - Q^*(t)$ denotes the relative complement of $Q^*(t)$ w.r.t. $\mathcal{A}^*(t)$. Then we claim,

Theorem 1: Let Q^* be an optimal policy for the optimization problem given in (3) when the data rates for all the users are i.i.d. in each timeslot. Then $\forall t = \{0, \cdots, N-1\}$, $Q^*(t) = argmax_{i\in A^*(t)}\{\mu_i(t)\}$ is the optimal opportunistic scheduling policy. We call this policy the Opportunistic Round Robin policy.

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 with 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 $\forall t$. ■

Now we relax the assumption that all the users are identical. But we assume that the user data rates are independent of each other and also across time. We claim that there exist $2^N - 2$ constants and an associated $argmax$ decision policy which is an optimal opportunistic scheduling policy. For simplicity of the arguments we consider some auxiliary notation as follows.

Let $Q^*_{\mathcal{A}}$ denote an optimal policy for a set of $\mathcal{A}$ users. Speci£cally by this we mean that in the next $|\mathcal{A}|$ timeslots each user from the set $\mathcal{A}$ gets service for one timeslot. Also denote the throughput associated with this optimal policy by $V^*_{\mathcal{A}}$. We show that there exists an optimal opportunistic scheduling policy by constructing one such policy.

Theorem 2: If the data rate of all the users are independent of each other and across the time then

$$\begin{gathered} Q^*(t) = Q^*_{\mathcal{A}^*(t)}(t) = argmax_i\{\mu_i(t) + V^*_{\mathcal{A}^*(t)-\{i\}}\} \\ A^*(0) = \mathcal{N}, A^*(t) = A^*(t-1) - Q^*(t-1) \end{gathered} \tag{4}$$

is the optimal opportunistic scheduling policy for the optimization problem given in (3).

As explained previously the constants $V^*_{\mathcal{A}^*(t)-\{i\}}$ are the throughput values of the optimum scheduling policy for the set of $\mathcal{A}^*(t) - \{i\}$ users that satis£es the short term fairness constraint. (Note that these constants are not similar to the constants λ_i. Only one λ_i constant is associated with a user i in the optimal long term policy, while $V_{\mathcal{A}^*(t)-\{i\}}$ depend on the user, the time index, and $\mathcal{A}^*(t)$. There are more than one such constants associated with each user.)

Proof: Clearly if $\mathcal{N} = \{i\}$, *i.e.*, for a singleton set the policy Q_i^* is trivially optimal. Thus $V_i^* = E(\mu_i)$. Now the optimality follows from an induction proof. Suppose for the set of users $\mathcal{N}$ the policy Q^* is optimal. If we add another user then the optimal policy would be to choose a user in time slot 0 optimally and then have an 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 as the recursive de£nition of set $\mathcal{A}^*(t)$ holds. The constants are the expected throughput values of the optimal policies for sets of type $\mathcal{A}^*(t) - i$. 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 $\mathcal{N}$). ■

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.

IV. Heuristic Policy

Until now we have analyzed three special cases of the opportunistic scheduling problem. The long term optimal policy selects a user having maximum "$\mu_i(t) + \lambda_i(t)$" in each timeslot while the £rst (respectively second) special short-term optimal policy selects a user with 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 de£ne the following heuristic policy for the general opportunistic scheduling problem de£ned in (1). The Heuristic Policy (HP) with a STF window of size M is de£ned as follows.

$$\forall t = \{0, \cdots, M-1\}, \;\; Q(t) = argmax_{i \in \mathcal{A}(t)}\{\mu_i + \lambda_i\} \quad (5)$$

Where, λ_i's are the constants from the LT policy. And,

$$A(0) = \mathcal{N}, \; N_i(0) = 0$$

And $N_i(t), \;\; A(t)$ are de£ned recursively as follows.

$$\begin{aligned} N_i(t) &= N_i(t-1) + I_{Q(t-1)=i} \\ A(t) &= A(t-1) - Q(t-1) I_{N_i(t)=M\phi_i} \end{aligned}$$

The Heuristic Policy can be outlined in 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 frame 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.

V. Simulation Results

In this section 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. The data rate for each HDR user is determined by the Signal to Noise Ratio (SNR) as shown in Table I taken from [9].

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

We use a stochastic approximation method as outlined in [5] to calculate the values of the constants λ_i's. Speci£cally we start with $\forall i, \;\; \lambda_i = 0$. Then in each timeslot we modify the λ_i values as follows:

$$\lambda_i(t+1) = \lambda(t) + (I_{Q_{LT}(t)=i} - \phi_i)\delta(t)$$

Where, $\delta(t)$ is a step-size function proportional to $1/t$. (This is a standard method to choose step-size in stochastic approximation algorithms.) The implementation of HP simulates the long term optimal policy to calculate the constants λ_i's on the ¤y; however the actual decision to select a user for service in every timeslot is done according to (5). (We have simulated a heuristic similar to the HP but with $\forall i, \lambda_i = 0$ 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 £rst 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 normal random variables (rv) with mean 0 and standard deviation 5 (for more details please refer to [5], [9]). 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 3. The (weighted) Round Robin has very low throughput (720 kbps in this scenario) compared to other policies hence we shall not consider it further. We make the following observations. The throughput

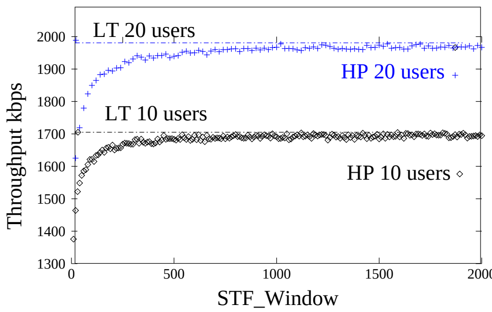

Fig. 3. Throughput vs STF window for different policies for i.i.d. HDR users, no. of timeslots = 10,000.

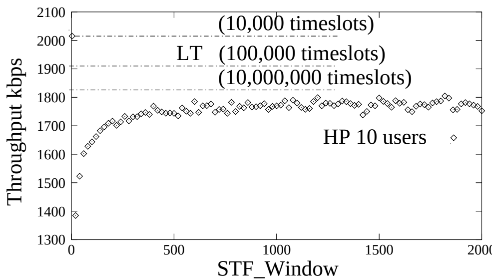

Fig. 4. Throughput vs STF window for different policies. 10 HDR users with different channel conditions and different ϕ_i's, no. of timeslots = 10,000.

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 in realistic channel conditions, we consider a case when the users have different channel distributions and different ϕ_i's. There are 10 users (see Table II) in the system.

TABLE II
USER DETAILS FOR SECOND SIMULATION SCENARIO

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

The channel SNR for each user is modeled as an autoregressive normally distributed channel. Speci£cally,

$$s_i(t+1) = \gamma s_i(t) + (1-\gamma) n_i(t+1)$$

Where $s_i(t)$ denotes the channel SNR in timeslot t for user i and $n_i(t)$ denotes the channel variations (noise terms) which are normally distributed independent rv's. These $n_i(t)$ rv's

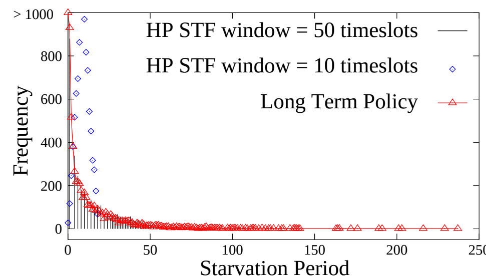

Fig. 5. Frequency distribution of starvation period for 10 HDR users with different channel conditions and different ϕ_i's, no. of timeslots = 10,000.

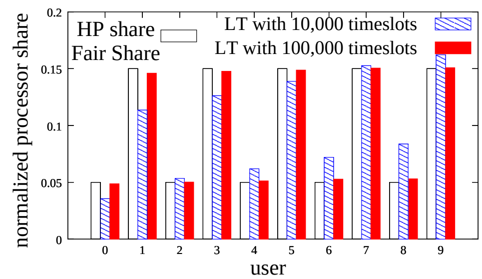

Fig. 6. Normalized processor share given to users by different policies. 10 HDR users with different channel conditions and different ϕ_i's

have a standard deviation of 15 and their mean values are shown in Table II. Users $0, 1$ have mean n_i of -4, users $2, 3$ have mean n_i of -2, and so on. γ, the auto-regression coef£cient is set to 0.7. The users with the same n_i mean 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 4. Figure 5 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 800 versus x axis value of 30 would mean that all users cumulatively have experienced starvation period of 30 for 800 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 5 we notice that HP is better than the LT policy in terms of short-term fairness. The 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 4, 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 6). 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 in£nite number of timeslots) this unfair behavior at the initial stage leads to unfair behavior of the long term policy over a £nite number of timeslots. This also explains the large difference between the throughput of the HP and the long term optimal policy in Figure 4 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 larger M. In real systems we expect that the users have £nite 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 £nite 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.

VI. Conclusions

In this paper we studied an opportunistic scheduling problem with strict short-term fairness constraints. The short-term fairness constraints require each user to be given its fair share over a speci£ed short term fairness window. The opportunistic scheduling problem with short term fairness constraints is a complex optimization problem and the general optimal policy is dif£cult to obtain. Hence we consider some special cases of the opportunistic scheduling problem and present optimal scheduling policies for these special cases. Motivated by the $argmax$ form of these optimal scheduling policies we then present a heuristic scheduling policy for the opportunistic scheduling problem with short term fairness constraints. We showed that the heuristic policy gives better control over short term fairness and the starvation period than the optimal long term policy which may not be fair in practical systems. Hence we conclude that our heuristic policy attains a good trade-off by guaranteeing short term fairness while achieving high average system throughput.

Acknowledgments

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

References

[1] P. Viswanath, D. N. C. Tse, and V. Anantharam, “Asymptotically optimal water£lling in vector multiple access channels,” *IEEE Transctions on Information Theory*, vol. 47, January 2001.

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

[3] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijaykumar, , and P. Whiting, “Providing a quality of service over a shared wireless link,” in *IEEE Communications Magazine*, vol. 39, February 2001, pp. 150–153.

[4] S. Shakkottai and A. Stolyar, “Scheduling algorithms for a mixture of real-time and non-real-time data in HDR,” in 17^{th} *International Teletraf£c Congress*, Septmber 2001.

[5] X. Liu, E. K. P. Chong, and N. B. Shroff, “Opportunistic transmission scheduling with resource-sharing constraints in wireless networks,” in *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 2053-2064, Oct 2001, p. 2001.

[6] S. S. Kulkarni and C. Rosenberg, “Opportunistic scheduling for wireless systems with multiple interfaces and multiple constraints,” To Appear in Proceedings of ACM/SIGCOM MSWiM, September 2003.

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

[8] T. Ng, I. Stoica, and H. Zhang, “Packet fair queuing algorithms for wireless networks with location dependent errors,” in *Proceedings of IEEE INFOCOM*, 1998.

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