EXTENSIONS OF PROPORTIONAL-FAIR SHARING ALGORITHMS FOR MULTI-ACCESS CONTROL OF MOBILE COMMUNICATIONS: CONSTRAINTS, FINITE QUEUES AND BURSTY DATA PROCESSES

Harold J. Kushner

Applied Mathematics Dept., Brown University, Providence RI 02912, USA, hjk@dam.brown.edu

Abstract We are concerned with the scheduling decisions (allocation of transmitter time, bandwidth and power) for multi-access mobile communications for data communications when the channels are randomly time varying. Time is divided into small scheduling intervals, called slots, and information on the channel rates for the various users is available at the start of the slot, when the user selections are made. There is a conflict between selecting the user that can get the most immediate data through and helping users with poor average throughputs. The Proportional Fair Sharing method (PFS) deals with such conflicts. In [4], [6] the convergence and basic qualitative properties were analyzed via stochastic approximation methods. The paths of the (suitably interpolated) throughputs converge to the solution of an ODE, akin to a mean flow. The behavior of the ODE completely describes the behavior of PFS. It has a unique equilibrium point that is asymptotically stable and optimal for PFS in that it is the maximizer of a concave utility function. There is a large family of such algorithms, each member corresponding to a concave utility function. Most past work assumed an infinite backlog of data. In many applications, the data arrival process for some users is bursty and data is queued until transmission, there might be minimal throughput constraints, or a balance between queue length (or delay) and throughput sought. The fact that some queues might be empty at times raises new issues. Natural modifications of PFS for these cases are shown to have the same properties. Simulations illustrate many of the unique features and the tradeoffs that are possible,

Keywords: Proportional fair sharing, throughput constraints, delay constraints, multi-access control, time varying channels

I. Introduction

Consider a multiaccess system for mobile data communications, with N mobiles and (for simplicity here only) a single base station. The rates of transmission for each channel are randomly varying. Time is divided into small intervals, called slots. In each slot one user is chosen. Estimates of the possible channel rates are available at the start of each slot, via estimates of the SNR (signal to noise ratio), obtained by use of a pilot signal. The user selection is based on a balance between the current possible channel rates and "fairness." The PFS (proportional fair sharing) algorithm makes the selection by comparing the channel rate for each user with its average throughput to date [1], [2], [3].

The evolution of the throughput under PFS can be written in a recursive manner, in the form of a stochastic approximation (SA) [7], and SA methods used for the analysis. The papers [4], [6] dealt with the asymptotic and qualitative properties via SA theory. The sequence of suitably interpolated throughputs converges to the solution of a "mean" ODE that completely characterizes the behavior of PFS. It has a unique and asymptotically stable equilibrium point $\bar{\theta}$ to which the throughputs converge. This in turn, gives other properties (e.g,, quantifying the scheduling gains over TDMA). The algorithm was shown to be optimal in that it gives the largest long term value of an associated utility function. In analyses of PFS, it is commonly assumed that there is an infinite backlog of data. Reference [6] also contained results when there are explicit bursty data arrival processes, with data queued in a finite buffer until transmitted, and similar results were obtained. Some extensions are in [5]; multiple simultaneous assignments, nonlinear dependence of channel rate on power, and the possibility that after a selection, a random a number of slots will be required.

This paper continues the work, emphasizing cases where some users are queueing bursty data arrival streams. If a user has an infinite backlog of data, then queueing delay is meaningless, but we might be concerned with minimum throughput guarantees. For users with queued bursty arrival streams, we suppose that queueing delays are important. So it is natural to be concerned with forms of PFS that allow reasonable tradeoffs among delay and throughput.

The way that fairness was defined in the original PFS concept together with the interpretation in terms of a utility function can be exploited to obtain natural PFS algorithms in such cases, with all of the analytical results carrying over. Section 2 reviews the past work. Section 3 concerns extensions to bursty data, minimal throughput constraints, and balances between throughput and queueing delay. Simulation data is presented in Sections 4–6. These represent only a sampling of the possibilities. They well illustrate the behavior and sensitivity to the data structure in these cases, and provide an overview of the possible tradeoffs among the quantities of interest.

II. PFS Algorithms and Past Results

The original PFS algorithm: Infinite data backlog. This section reviews some results for the original PFS algorithm. There are N users, each with infinite backlog, sending data to mobiles via a single transmitter, with the rates or capacity of each channel varying randomly. One user is scheduled at a time. If user i is selected in slot n, it transmits $r_{i,n}$ packets, where $\{r_{i,n}, n < \infty\}$ is a random sequence. The end of slot n is called *time* n. Let $I_{i,n}$ be the indicator of the event that user i chosen for slot n. One definition of throughput up to time n for user i is

$$\theta_{i,n} = \sum_{l=1}^{n} r_{i,l} I_{i,l}/n, \qquad (2.1)$$

which can be written in recursive form as

$$\theta_{i,n+1} = \theta_{i,n} + \epsilon_n \left[I_{i,n+1} r_{i,n+1} - \theta_{i,n} \right], \quad \epsilon_n = 1/(n+1). \qquad (2.2)$$

A useful alternative definition of throughput discounts past values as, for a small discount factor $\epsilon > 0$,

$$\theta^{\epsilon}_{i,n} = (1-\epsilon)^n \theta^{\epsilon}_{i,0} + \epsilon \sum_{l=1}^{n} (1-\epsilon)^{n-l} r_{i,l} I_{i,l}, \qquad (2.3)$$

where θ^{ϵ}_n = discounted throughput at time n. (2.3) is

$$\theta^{\epsilon}_{i,n+1} = \theta^{\epsilon}_{i,n} + \epsilon \left[I_{i,n+1} r_{i,n+1} - \theta^{\epsilon}_{i,n} \right]. \qquad (2.4)$$

Let $d_i > 0$, which can be as small as desired. For positive weights w_i, the original PFS algorithm chooses the user at time n maximizing in

$$\arg\max\{i : w_i r_{i,n+1}/[\theta^{\epsilon}_{i,n} + d_i], i \le N\} \qquad (2.5)$$

Algorithms (2.2) and (2.4) are of the SA form [7], and the results of SA were used for their analysis. The approach is able to handle quite general channel rate processes and also arbitrary data arrival processes. The basic algorithm and results are typical of a large family, indexed by some concave utility function, and which is actually maximized by the associated algorithm. The same advantage holds for the extensions in Section 3. The treatment of (2.2) and (2.4) differs mainly in notation.

The assumptions to follow are special cases of the more general forms in [6], but are sufficient for our purposes. A2.3 is used to avoid degeneracy and is unrestrictive. It holds if when a component θ_i is small enough then there is a nonzero chance that user i will be chosen, no matter what the other components are. Define $\theta_n = \{\theta_{i,n}, i \le N\}$, $\theta^{\epsilon}_n = \{\theta^{\epsilon}_{i,n}, i \le N\}$, and $R_n = \{r_{i,n}, i \le N\}$. We will abuse terminology by using θ_i and θ_j for the i-th and j-th components, resp., of the vector θ. The assumptions all hold for Rayleigh fading.

A2.1. R_n is stationary, and has a density $P_r(\cdot)$, with bounded variance.

A2.2. The distribution of R_{n+1}, given $R_l, l \le n$, has density $p_n(\cdot)$ and these conditional densities are uniformly absolutely continuous with respect to Lebesgue measure. The density of R_{n+m}, given $\{R_l, l \le n\}$, converges to $P_r(\cdot)$ as $m \to \infty$.

Let $I_i(R,\theta)$ denote the indicator of the event that user i is selected when the channel rate and throughput vectors are R, θ, resp. Define (stationary expectation)

$$\bar{h}_i(\theta) = E r_i I_i(R,\theta). \qquad (2.6)$$

A2.3. There are small positive δ, δ_1 such that for any i $\bar{h}_i(\theta) \ge \delta_1$, if $\theta_i \le \delta$.

Bursty inputs and bounded queues. The above formulation was for the original case where each user has an infinite backlog of data. In many applications the data for some users arrives at random and is queued until transmitted. Now suppose that some users have an infinite backlog and others bursty arrivals that are queued in a finite buffer. Overflow packets are lost. This was treated in [6] and the following is a special case of the assumptions there, but which is adequate for our purposes.

A2.4. If user i has bursty arrival data, its arrival process is compound Poisson with bounded batches, with probability p_i of an arrival to queue i in any slot, and where the sequences of batches are independent and identically distributed, and independent of the channel rate processes.

Let $L_{i,n}$ denote the content of queue i at time n and define the mean queue level

$$X_{i,n} = \sum_{l=1}^{n} L_{i,l}/n, \qquad (2.7)$$

or, in recursive form,

$$X_{i,n+1} = X_{i,n} + \epsilon_n \left[L_{i,n+1} - X_{i,n} \right]. \qquad (2.8)$$

If user i has infinite backlog, set $L_{i,n} = \infty$. For the case (2.4), use $\epsilon_n = \epsilon$. The decision rule is now

$$\arg\max\{i : w_i \min(r_{i,n+1}, L_{i,n})/[\theta^{\epsilon}_{i,n} + d_i], i \le N\}. \qquad (2.9)$$

If the decision maker does not know the current queue size, only if it is empty or not, one might use (2.5), but it will be less efficient. The average throughput is now

$$\theta_{i,n} = \sum_{l=1}^{n} \min(r_{i,l}, L_{i,l-1}) I_{i,l}/n. \qquad (2.10)$$

Discretized rates. It is often the case that the possible rates of transmission belong to some discrete set. Suppose that the decisions are made via (2.9), but once the choice is made, the actual transmission is the discretization of this rate to some finite set, with the discretization used in computing the throughputs (2.2)–(2.4) or (2.10). Then all of the following results continue to hold.

A problem that arises in the bursty-arrival finite-buffer case concerns empty queues. A queue might be empty for a relatively long period. If the queue length is important owing to its connection with delay, we might be more concerned with mean delay only when there is queued data, as opposed to the average over all time. Then one might modify (2.7) and (2.10) by dividing only by the number of times that the queue is not empty and not by n. For simplicity we stick to definitions (2.7), (2.10).

For the present case, let $I_i(R, \theta, L)$ denote the indicator that user i is selected when the channel rate, throughput and queue length vectors are R, θ, L, resp. Define

$$\bar{h}_i(\theta) = E \min(r_i, L_i) I_i(R, \theta, L) \tag{2.11}$$

Convergence and optimality results. The usual SA analysis uses continuous time interpolations. Define $\theta^\epsilon(\cdot)$ by $\theta^\epsilon(t) = \theta^\epsilon_n$ for $t \in [n\epsilon, n\epsilon + \epsilon)$. Define the continuous time interpolation $\tilde{\theta}(\cdot)$ of θ_n in (2.2) analogously, with intervals ϵ_n in lieu of the constant ϵ. The next theorem says that the limit points of (2.2) and (2.4) are contained in those of the ODE (2.12).

Theorem 1. ([7, Theorems 2.2 and 2.3, Section 8.2].) *Assume algorithm* (2.4), (2.9) *and* A2.1–A2.2, A2.4. *Then, as $\epsilon \to 0$, $\theta^\epsilon(\cdot)$ converges to the solution of*

$$\dot{\theta}_i = \bar{h}_i(\theta) - \theta_i, \quad i \le N. \tag{2.12}$$

The same result hold for $\tilde{\theta}(t + \cdot)$ as $t \to \infty$.

Limit points of PFS. The ODE (2.12) has what is called the “cooperative” property. This fundamental property says simply the following: If θ_i is increased, then the other users are not less likely to be chosen. More generally, an ODE $\dot{x} = f(x)$ is said to be cooperative if for any i and vectors x, y with $x \le y$ and $x_i = y_i$, we have $f_i(x) \le f_i(y)$. The property implies the following important monotonicity theorem [8, Proposition 1.1]: Consider vectors θ^0, θ^1, with all components of θ^1 being no smaller than those of θ^0. Then the components of the solution starting at θ^1 are no smaller than those corresponding to the solution starting at θ^0. The proofs in [4], [6] of Theorems 2 and 3 use the monotonicity property together with the strict concavity of the log function, and both properties are essential for the extensions. The next result says that the throughputs converge to unique limiting values

Theorem 2 *Assume rule* ((2.4), (2.9)) and A2.1–A2.4. *The ODE* (2.12) *is cooperative and the limit point $\bar{\theta}$ is unique and asymptotically stable. $\theta^\epsilon_n \to \bar{\theta}$ as $\epsilon \to 0$ and $\epsilon n \to \infty$. For* (2.2), *$\theta_n \to \bar{\theta}$ as well, as $n \to \infty$.*

Maximizing a utility function. PFS is also optimal, as follows. The PFS rule of (2.9) is a type of “ascent” algorithm for the strictly concave utility function

$$U(\theta) = \sum_i \log(d_i + \theta_i). \tag{2.13}$$

By a first order Taylor expansion,

$$U(\theta_{n+1}) - U(\theta_n) = \\ \epsilon_n \sum_i [\min(r_{i,n+1}, L_{i.n}) I_{i,n+1} - \theta_{i,n}]/[d_i + \theta_{i,n}] + O(\epsilon_n^2). \tag{2.14}$$

Since $\sum_i I_{i,n+1} = 1$, to maximize the first order term one must choose $I_{i,n+1}$ by (2.9). Theorems 1–3 can be extended to algorithms based on any strictly concave utility function, not just (2.13). This yields a family of algorithms that allow different tradeoffs between the values of the current channel rates and throughputs [6]. The forms of the extensions are motivated by the maximization of the first order term in such expansions. Theorem 3 asserts that the rule (2.9) maximizes $U(\cdot)$ [6]. The Appendix contains some details of the proof.

Theorem 3. *Under* A2.1–A2.4, $\lim_{n\to\infty} U(\theta_n)$ *and* $\lim_{\epsilon\to 0, n\to\infty} U(\theta^\epsilon_n)$ *are maximized over all feasible alternatives.*

III. Extensions

Minimum throughput constraints for some users with infinite backlog. Now let there be constraints on the minimum throughput in that we desire $\theta_i \ge a_i$. Set $a_i = 0$ if no minimum is desired. If the current θ_i value is less than a_i then user i needs to be given some advantage in that it should be selected even for some channel rates that are lower than what (2.9) would require. But how much advantage ? Overall efficiency is still a concern, and user i should not be selected if its current channel rate is too low. This reasoning (see also the discussion in Section 4) argues against the use of hard constraints. We use the following “soft” approach. For each i, let $q_i(\theta_i)$ be a differentiable penalty function which is zero if $\theta_i \ge a_i$, negative for $\theta_i < a_i$, and whose derivative $q_{i,\theta_i}(\theta_i)$ is non decreasing in θ_i as θ_i decreases. For example, $q_i(\theta_i) = K_i \max\{0, |\theta_i - a_i|(\theta_i - a_i)\}, K_i > 0$. Analogously to the role of (2.14) in getting (2.9), let the decision rule maximize the first order term in the expansion of $U(\theta_{n+1}) - U(\theta_n)$ where

$$U(\theta) = \sum [w_i \log(\theta_i + d_i) + q_i(\theta_i)], \tag{3.1}$$

a strictly concave function (or with another desired strictly concave function used in lieu of the log). This yields the rule

$$\arg\max_i \left\{ \min(r_{i,n+1}, L_{i,n}) \left[\frac{w_i}{\theta_{i,n} + d_i} + q_{i,\theta_i}(\theta_{i,n}) \right] \right\}, \tag{3.2}$$

where $q_{i,\theta_i}(\cdot)$ is the derivative of $q_i(\cdot)$ with respect to θ_i. Theorems 1–3 still hold under A2.1-A2.4. The ODE is (2.12) with the indicator function $I_i(R, \theta, L)$ being that arising from the decision rule (3.2).

The use of the penalty function allows flexibility near the boundaries a_i, since the choice will depend on both the current channel rates and by how much the constraint is violated. The use of penalty functions does not assure that the constraints will be satisfied. They might not be feasible. But it does give an advantage to users whose throughput is less than desired, and it puts a price on constraint violations.

Weighing both throughput and queue length. So far, we have been concerned with throughputs, with or without constraints on the minimal values. Some users whose input processs are bursty might be concerned with queueing delays as well as assuring that all the inputs are eventually transmitted. Let B denote the set of such users. This need can be accomodated by the use of the utility function, where the $H_i(\cdot)$ are strictly convex,

$$U(\theta, X) = \sum_i w_i \log(\theta_i + d_i) - \sum_{i \in B} H_i(X_i), \qquad (3.3)$$

The decision rule corresponding to (3.3) is

$$\arg\max_i \left\{ \min(r_{i,n+1}, L_{i,n}) \left[\frac{w_i}{\theta_i + d_i} + H_{i,X_i}(X_i) \right] \right\}. \qquad (3.4)$$

Let $I_i(R, \theta, L, X)$ denote the indicator that user i is selected when the channel rate, throughput, queue length, and sample mean queue length vectors are R, θ, L, X, resp. Then the ODEs for throughput and mean queue length are (with $\bar{a}_i$= mean input rate for user i)

$$\dot{\theta}_i = E \min\{r_i, L_i\} I_i(R, \theta, L, X) - \theta_i,$$
$$\dot{X}_i = EL_i - X_i.$$

Other extensions. These are only a few of the possible extensions. One can constrain minimum throughputs for users with infinite backlog and the maximum mean queue length for users with bursty inputs. One just subtracts appropriate strictly convex constraint functions from the utility function.

IV. Numerical Data: Minimum Throughput Constraints

The model data: The arrival and channel rate processes. A sampling of numerical data for two-user problems is discussed in this and in the next two sections. This section deals with minimum throughput constraints on the user with infinite backlog. Section 5 deals with classical PFS with varying weights, when both users have bursty input processes. In Section 6, both queues have bursty inputs, and the queue length is also weighted in the decision, as in (3.3), (3.4).

There are two classes of users, those with infinite backlog and those with bursty arrival processes. In all bursty arrival cases, the input processes were compound Poisson, with rate 0.1 and the batches uniformly distributed in [0,20], so that average arrival rate/slot is 1.0. Using smaller arrival probabilities and larger batch sizes (but with the same mean value) led to similar results, but with the mean queue levels larger, being a little less than proportional to the batch size. The channel rate processes were mutually independent. Two types of channel rate processes were used. For the first, the channel rates were mutually independent and took values in the interval $[0, A]$. This is referred to as the iid$[0, A]$ case. The value of A was either 5 or 7.5, yielding mean channel rates/user that would be either 2.5/2 or 3.25/2, if the channel was equally divided between the two users. The other channel rate process was a simple first order Markov process, defined by $r_{i,n+1} = .8r_{i,n} + \xi_{i,n+1}$, where the ξ were uniformly distributed on either [0,1] or [0,1.5]. These have the same means as the two iid cases, and are referred to as correl$[0, A]$, with A being either 1 or 1.5. Similar conclusions hold for more complex correlations. In all cases the listed values of θ_i, X_i were the averages of the end values of several very long runs.

The main issues are the possible tradeoffs between the quantities of interest, the θ_i and X_i, and the price to be paid on the others for a given improvement in one of them. The tables illustrate the possibilities.

Minimum throughput constraints. Consider the first model of Section 3, with user 1 having an infinite backlog, and user 2 bursty arrivals. There is a minimum throughput constraint for user 1, with constraint function $q_1(\theta_1) = (2K/3)([a_1 - \theta_1]^+)^{3/2}$. Other strictly convex constraint functions gave results similar to those below.

Consider Table 4.1. The first row is just unconstrained PFS. The θ_2 are all unity since all of users 2's inputs are eventually transmitted. In all cases the sums of the throughputs is greater than 2.5, the average channel rate/slot over the two channels. The value $K = 0.5$ achieves the minimal desired throughput (for about a 10 percent increase in θ_1 over the first line, but at the cost of nearly doubling X_2. If K is increased to 1.0, X_2 more than doubles again. With the larger K, user 1 effectively becomes "more aggressive," and is more likely to be selected when its throughput is less than the desired value. The average throughput increases as K increases, since all of user 1's data are transmitted. The value of X_2 will become infinite if K increases much beyond $K = 1$. These facts argue against using a hard constraint, under which user 1 is always selected if its current throughput is less than the desired level. They also underline the importance of understanding the possible tradeoffs, so that an excessive value of K is not used. The numbers are better than they seem, if we keep in mind that the queue for user 2 is often empty or small. Then user 1 is always selected, whether its current channel rate is good or not.

Table 4.1. Min. Constr. on user 1				
iid[0,5], user 2-bursty input				
K	a_1	θ_1	θ_2	X_2
0.0	n.a.	1.79	1	6.23
0.2	2	1.88	1	7.55
0.5	2	2	1	12.33
1	2	2	1	26.5
2	2	2	1	132

Table 4.2. Min. Constr. on user 1				
correl[0,1], user 2-bursty input				
K	a_1	θ_1	θ_2	X_2
0.0	n.a.	1.5	1	6.1
0.2	2	1.6	1	10.44
0.3	2	1.64	1	16.9
0.5	2	1.75	1	699
1	2	1.97	.75	∞

Table 4.3. Min. Constr. on user 1				
iid[0, 7.5], user 2-bursty input				
K	a_1	θ_1	θ_2	X_2
0.0	n.a.	2.88	1	3.29
0.5	3.25	3.18	1	4.87
1	3.25	3.25	1	8.66
1.5	3.25	3.25	1	13.74

Comparing Table 4.1 with 4.2 (which has the same mean channel rates), we see that the throughputs decrease and the unstable point for user 2 is reached earlier as the correlation of the channel rate process increases. This seems to be the general case; correlation in the channel rate process hurts throughputs and queue lengths. Under correlation, the rate process has less variability so that it is harder to exploit variations in the rates to improve the operation. Note that for $K = 1$, the queue for user 2 is unstable, and only .97 of its inputs are transmitted.

Table 4.3 is for the iid channel rate process, but with a mean channel rate of 3.25, a fifty percent increase over that of Table 4.1. Owing to the greater excess capacity, it is easier to satisfy the constraint and, at best, we have a total throughput of 4.25, compared to the channel average of 3.25. A heavy penalty (in terms of X_2) is paid for assuring a throughput $\theta_1 = 3.18$ vs $\theta_1 = 3.25$. Indeed, one must always take care to understand the cost of reducing one variable in terms of the increase in other variables.

V. Classical PFS, Both With Bursty Inputs, and With Varying Weights

In this section both users having bursty inputs. Tables 5.1 and 5.2 present an alternative way of getting tradeoffs between queue lengths, hence between delays. By varying the w_i in (2.9), one can graph the set of possible pairs X_1, X_2. The queues are longer in the correlated channel rate case, as is the sensitivity of the length of the queue with the unit weight to the value of the one with the larger weight, possibly due to the fact that once a user is selected, it is more than likely to be selected in the next few slots as well. In addition, there is less "effective" variability of the channel rates in the correlated case, thereby reducing the opportunities.

Table 5.1. PFS		
iid[0, 5], Bursty inputs		
w_1, w_2	X_1	X_2
1, 1	7.3	7.3
1, 2	11.4	5.5
1, 3	14.2	5.2

Table 5.2. PFS		
correl[0,1], Bursty inputs.		
w_1, w_2	X_1	X_2
1, 1	10.6	10.6
1, 1.5	20.8	5.6
1, 2	23	5
1, 3	25.1	4.86

VI. PFS with Varying Weights on Queue Length

In this example, the delay for user 2 is more important than that of user 1, and this is treated by putting an explicit penalty on user 2's queue length. It is an alternative to the approach of the last section. In Tables 6.1–6.4, the utility function and decision rule are (3.3) and (3.4), resp. Both users have bursty input processes, but there is a penalty on the queue length of user 2, with $H_1(X_1) = 0, H_2(X_2) = (2K/3)|X_2|^{3/2}$. The results were similar with other penalty functions.

Table 6.1.			
iid,[0 5], mean/user= 2.5,			
$w_1,\ w_2$	K	X_1	X_2
1, 1	0, 0	7.3	7.3
1, 1	0, .5	12.2	5.45
1, 1	0, 1	15.1	5.13
1, 1	0, 2	18.94	5

Table 6.2.			
correl =.8, mean/user= 2.5,			
$w_1,\ w_2$	K	X_1	X_2
1, 1	0, 0	10.6	10.6
1, 1	0, .3	22	5.17
1, 1	0, .5	24.3	5
1, 1	0, 1	25.4	4.84
1, 1	0, 2	26.6	4.83

Table 6.3.			
iid,[0 7.5], mean/user= 3.25			
$w_1,\ w_2$	K	X_1	X_2
1, 1	0, .3	4.2	3.3
1, 1	0, .5	4.46	3.2
1, 1	0, 1.5	5.2	3.1
1, 1	0, 10	6.453	3.02

Table 6.4			
correl=.8, mean/user= 3.25			
w_1, w_2	K	X_1	X_2
1, 1	0, .1	4.66	3.5
1, 1	0, .3	5.4	3.1
1, 1	0, 1	6	2.9
1, 1	0, 5	6.3	2.86

By graphing the pairs X_1, X_2 as the weight K varies, we can get a good understanding of the possible tradeoffs and the implicit price associated with a decrease in X_2. As K increases, in all cases the marginal improvement in X_2 goes to zero, and the ratio of the improvement in X_2 to the increase in X_1 goes to infinity rapidly, a fact that is not obvious a priori. This holds no matter what the excess capacity is. So, the choice of weight can be a delicate matter. The behavior of the correlated channel rate case is similar to that in Section 5 and probably for the same reason.

VII. Appendix: Some details of the proof for Minimum Throughput Constraints

The proofs of the extensions involve little change from those in [6], and there is little space for development. We will comment briefly on one part of the proof of Theorem 3 for the minimum throughput constraints for the infinite backlog case. For simplicity in notation, we work with (2.4). The proof in [6] first used the monotonicity property and the strict concavity of the utility function to show that all paths of the solution $\theta(\cdot)$ to (2.12) end up at a unique stable point $\bar{\theta}$, which is also the limit of the throughput processes under PFS. Define the set $Q(\bar{\theta}) = \{\theta : \theta_i \geq \bar{\theta}_i, i \leq N\}$. Here we use the utility function (3.1). For notational simplicity we use the throughput definition (2.4). Suppose that there is some assignment $\hat{I}_{i,n}, i \leq N$, under which the average throughput converges to a vector $\hat{\theta}$, where $U(\hat{\theta}) > U(\bar{\theta})$. Thus $\hat{\theta}$ must be in $Q(\bar{\theta})$.

Now, consider the algorithm, started at $\bar{\theta}$, but with the alternative assignment rule $\hat{I}_{i,n}, i \leq N$, used. The $I_{i,n}, i \leq N$, still denotes the assignment given by (2.9) at whatever the current value of θ^ϵ_n (determined by the policy $\hat{I}_{i,n}, i \leq N$) is. Then, modulo an error of order $O(\epsilon)t$ that is due to the first term in the Taylor expansion only being taken, the maximization in (3.2) yields

$$\begin{aligned} &U(\theta^\epsilon(t)) - U(\bar{\theta}) \\ &= \epsilon \sum_i \sum_{l=0}^{[t/\epsilon]-1} \left[\frac{w_i}{\theta^\epsilon_{i,n} + d_i} + q_{i,\theta_i}(\theta^\epsilon_{i,l}) \right] \left(r_{i,l+1}\hat{I}_{i,l+1} - \theta^\epsilon_{i,l} \right) \\ &\leq \epsilon \sum_i \sum_{l=0}^{[t/\epsilon]-1} \left[\frac{w_i}{\theta^\epsilon_{i,n} + d_i} + q_{i,\theta_i}(\theta^\epsilon_{i,l}) \right] \left(r_{i,l+1} I_{i,l+1} - \theta^\epsilon_{i,l} \right) \end{aligned} \tag{7.1}$$

where $[t/\epsilon]$ = integer part of t/ϵ.

The convergence of the throughput to $\hat{\theta}$ under $\hat{I}_{i,n}, i \leq N$, together with (7.1), implies that as $\epsilon \to 0$ (hence $\theta^\epsilon(\cdot) \to \theta(\cdot)$, a continuous function)

$$\begin{aligned} &U(\theta(t)) - U(\bar{\theta}) \\ &= \int_0^t \sum_i \left[\frac{w_i}{d_i + \theta_i(s)} + q_{i,\theta_i}(\theta_i(s)) \right] \left(\hat{\theta} - \theta_i(s) \right) ds \\ &\leq \int_0^t \sum_i \left[\frac{w_i}{d_i + \theta_i(s)} + q_{i,\theta_i}(\theta_i(s)) \right] \left(\bar{h}_i(\theta(s)) - \theta_i(s) \right) ds. \end{aligned}$$

This and the fact that $\bar{\theta}$ is an equilibrium point of the ODE (2.12) implies that

$$\begin{aligned} \dot{U}(\theta(t))\big|_{t=0} &= \sum_i \left[\frac{w_i}{d_i + \bar{\theta}_i} + q_{i,\theta_i}(\bar{\theta}_i) \right] \left(\hat{\theta}_i - \bar{\theta}_i \right) \\ &\leq \sum_i \left[\frac{w_i}{d_i + \bar{\theta}_i} + q_{i,\theta_i}(\bar{\theta}_i) \right] \left(\bar{h}_i(\bar{\theta}) - \bar{\theta}_i \right) = 0. \end{aligned}$$

Since the term on the right of the first line is positive, there is a contradiction to the existence of $\hat{\theta}$. Consequently, PFS gives an optimal assignment rule for the given utility function.

Conclusions. We obtain effective PFS algorithms for conditions of bursty data, minimal throughput constraints, and where a balance between throughput and queueing delay is required. The analytical results concerning convergence and qualitative properties for the classical cases carry over. Simulations illustrate the behavior and sensitivity to the data structure in these cases, and provide an overview of the possible tradeoffs among the quantities of interest.

References

[1] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushyana, and S. Viterbi. CDMA/HDR: a bandwidth efficient high speed wireless data service for nomadic users. *IEEE Comm. Magazine*, 38:70–77, 2000.

[2] S. Borst. User level aware performance of channel-aware scheduling algorithms in wireless data networks. In *Proc. Infocom, San Francisco*, New York, 2002. IEEE Press.

[3] J. M. Holtzman. Asymptotic analysis of proportional fair sharing. In *Proc. Personal, Indoor, and Mobile Radio Communications, Vol. 2*, pages 33–37, New York, 2001. IEEE Press.

[4] H.J. Kushner and P.A. Whiting. Asymptotic properties of proportional-fair sharing algorithms. In *Proc., 2002 Allerton Conf.*, Champaigne-Urbana, IL, 2002. Univ. of Illinois Press.

[5] H.J. Kushner and P.A. Whiting. Asymptotic properties of proportional-fair sharing algorithms: Extensions of the algorithm. In *Proc., 2003 Allerton Conf.*, Champaigne-Urbana, IL, 2003. Univ. of Illinois Press.

[6] H.J. Kushner and P.A. Whiting. Convergence of proportional-fair sharing algorithms under general conditions. *IEEE Trans. Wireless Communications*, 3:1250–1259, 2004.

[7] H.J. Kushner and G. Yin. *Stochastic Approximation Algorithms and Applications.* Springer-Verlag, Berlin and New York, 1997. Second edition, 2003.

[8] H.L. Smith. *Monotone Dynamical Systems: An Introduction to Competitive and Cooperative Systems, AMS Math. Surveys and Monographs, Vol. 41.* American Mathematical Society, Providence RI, 1995.

Acknowledgements

This work was partially supported by ARO Contract DAAD-19-02-1-0425 and NSF Grant ECS 0097447.