Conference Proceedings

2006

Opportunistic Splitting Algorithms for Wireless Networks with Fairness Constraints

Xiangping Qin
xqin@siu.edu

Randall Berry
Northwestern University

Recommended Citation

Opportunistic Splitting Algorithms for Wireless Networks with Fairness Constraints

Xiangping Qin
Dept. of Electrical and Computer Engineering
Boston University
8 St. Mary's Street, Boston, MA 02215
Email: xqin@bu.edu

Randall Berry
Dept. of Electrical and Computer Engineering
Northwestern University
2145 Sheridan Rd., Evanston IL 60208
Email: rberry@ece.northwestern.edu

Abstract— In wireless networks, it is well established that the throughput can be increased by opportunistically scheduling transmissions to users that have good channel conditions. Several "opportunistic" medium access control protocols have been developed, which enable distributed users to opportunistically transmit without requiring a centralized scheduler. In this paper, we consider opportunistic splitting algorithms, where a sequence of mini-slots is used to determine the appropriate user to schedule at each time. In prior work, this type of algorithm has been developed for homogeneous systems in which all users have independent and identically distributed (i.i.d.) channel statistics. Here, we specify new splitting algorithms for a heterogeneous environment that may also include fairness constraints. The performance of the splitting algorithms are characterized via analysis and simulations. In particular, we show that in certain cases, a heterogeneous algorithm will perform at least as well as the homogeneous algorithm in a system with the same total number of users.

I. INTRODUCTION

It is well established that the throughput of wireless networks can be increased by opportunistically transmitting to users when they have good channel conditions. A variety of approaches for opportunistic scheduling have been studied such as [1]–[5], where a centralized controller decides which users to schedule given the channel gains of every user. In large networks, the overhead required for such a centralized approach can become prohibitive. To reduce this overhead for the uplink in a wireless network, various distributed opportunistic scheduling algorithms have been proposed, e.g. [6]–[10]. These algorithms assume that each transmitter has *distributed channel knowledge*, i.e., each transmitter knows its own channel gain, but has no knowledge of the other user's gains. For example, this could be estimated from a pilot signal broadcast by the receiver in a time-division duplex system. Given this distributed knowledge, the main goal is to determine which user should be scheduled without requiring all users to feedback their channel gains to a centralized controller.

In this paper, we consider a distributed scheduling algorithm based on the idea of an *opportunistic splitting algorithm* presented in [7]. In such an algorithm, a sequence of mini-slots is used to determine which user is scheduled. Each user

transmits in a given mini-slot if its channel gain lies in a given range. From mini-slot to mini-slot, the range is adjusted until only the user with the best channel gain is transmitting. In [7], an opportunistic splitting algorithm is presented for a homogeneous system, where every user's channel gains are i.i.d. In that case, it is shown that independent of the fading distribution and the number of users, the average number of mini-slots required to find the best user is bounded by 2.509. This suggests that overhead requirements of an opportunistic splitting algorithm will scale to accommodate a large number of users.

Here we present generalizations of the opportunistic splitting algorithm from [7], which can be applied to a heterogeneous network, where the channel gains of different users are still independent, but may not be identically distributed. Additionally, our algorithms can take into account fairness constraints based on the concept of *distribution fairness* introduced in [11]. This is a form of temporal fairness, where in the long run, each class of users is guaranteed to be scheduled a fixed fraction of time. We first consider an algorithm for a system where the number of users in each class involved in a collision is known. In this case we show that the average number of mini-slots required by this algorithm is no greater than that required in a homogeneous system with the same number of users, where the splitting algorithm from [7] is used. In particular, this means the average number of mini-slots is still upper-bounded by 2.509. We then consider an algorithm where the number of users involved in a collision is not known. In this case, the average number of mini-slots, until the first collision or success is less than that in a homogeneous system, but we can not guarantee that the average total number of mini-slots will be less. Simulation results suggest that the number average number of mini-slots increases, but only slightly. Some preliminary work along these lines is reported in [11], where we presented a different splitting algorithm for a heterogeneous network and only analyzed its performance for a two user network. In this paper, we give a different splitting algorithm and a more rigorous analysis that applies to an arbitrary number of users.

Next we give specify the system model and the definition of distribution fairness that we will use. In Section III-A, we give the first splitting algorithm for the case where the number

of users in each class involved in a collision is known. Next, in Section III-B, we give an algorithm for the case where the number involved in a collision is not known. Several proofs are given in the appendices.

II. MODEL DESCRIPTION AND DISTRIBUTION FAIRNESS

We consider the up-link in a wireless network with n users all transmitting to a common receiver. The channel between each user and the receiver is modeled as a time-slotted, block-fading channel. During each time-slot, the received signal $y(t)$ is given by

$$y(t) = \sum_{i=1} \sqrt{H_i}\, x_i(t) + z(t),$$

where $x_i(t)$ is the transmitted signal of user i, H_i is the user's channel gain, and $z(t)$ is additive white Gaussian noise. The channel gain is assumed to be fixed during each time-slot and to randomly vary between time-slots. The channel gains of each user in each time-slot are modeled as independent random variables, with continuous cumulative distribution functions (c.d.f.s) $F_i(h_i)$ on $[0, \infty)$ for all $i = 1, \ldots, n$. Let $\bar{F}_i(h_i) = 1 - F_i(h_i)$ denote complimentary c.d.f. for user i's channel gain.

We consider a time division multiplexed (TDM) system, where during each time-slot at most one user can successfully transmit. Users are assumed to be rate-adaptive, so that the transmission rate if user i successfully transmits in a time-slot is given by $R(h_i)$, where $R(\cdot)$ is an increasing function of the user's channel gain. We assume that all users are infinitely back-logged, and focus on the average throughput achieved by each user. Clearly in such a system, the total throughput will be maximized by scheduling the user with the best channel gain in each time-slot. It is well know that such a throughput maximizing approach can be unfair. Next, we review one the definition of fairness from [11]. For this purpose, consider a centralized TDM scheduler which, given the vector of channel gains $\mathbf{h} = (h_1, h_2, ..., h_n)$ at each time-slot, schedules one of the users to transmit. Let $A(\mathbf{h})$ denote a (stationary) scheduling policy, i.e., $A(\mathbf{h}) = i$ if user i is scheduled when the joint channel state is $\mathbf{h}$. Let $\{p_1, \ldots, p_n\}$ be a probability mass function (p.m.f.). A scheduling rule $A(\mathbf{H})$ is defined to be *distribution fair* with parameters $p_1, p_2, ..., p_n$, if it solves

$$\min_{A(\mathbf{H})} E_{\mathbf{H}} \left(\sum_{i=1} \bar{F}_i(H_i) 1_{A(\mathbf{H})=i} \right) \tag{1}$$

subject to: $\text{Prob}\{A(\mathbf{H}) = i\} = p_i$ for $i = 1, \ldots, n$.

Here, $1_{A(\mathbf{H})=i}$ is the indicator function for the event $A(\mathbf{H}) = i$. The p.m.f. $\{p_1, \ldots, p_n\}$ specifies the fraction of time that each user will transmit. Given this constraint, a distribution fair scheduler will attempt to schedule users with a small value of $\bar{F}_i(h_i)$; this corresponds to a user having a channel that is statistically strong relative to its own distribution.

The optimization problem in (1) is similar to one studied in [2], which considers maximizing the sum expected utility under the same constraints. Viewing the complimentary distribution function as a particular type of utility function, the
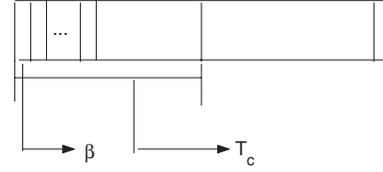
Fig. 1.   A time-slot made up of several mini-slots.

results in [2] can be applied to show that the optimal solution to (1) is given by

$$A(\mathbf{H}) = \arg \min_{i=1} (\bar{F}_i(H_i) + v_i), \tag{2}$$

where the parameters v_i for $i = 1, .., n$ are chosen to satisfy the constraints $\text{Prob}\{A(\mathbf{H}) = i\} = p_i$. If $p_i > p_j$, it can be shown that $v_i < v_j$. For convenience, we can set $v_j = 0$ for $i = \max_{i=1} p_i$, so that $v_j \geq 0$, for all j.

III. SPLITTING ALGORITHMS FOR HETEROGENEOUS NETWORKS

For a given vector of channel gains $\mathbf{h}$, a centralized distribution-fair scheduler schedules the user given by (2). In this section, we describe splitting algorithms with the goal of scheduling the same user in a distributed manner. At the beginning of each time-slot, the splitting algorithms use several mini-slots with length β to communicate with the base station (see Fig. 1). Here β is equal to the round-trip time required for a user to transmit a small reservation packet and detect if a collision occurs. Let T denote the length of one time-slot during which a user's channel gain is fixed (i.e., T is the length of a block in the block-fading model). We consider an idealized model where each time-slot contains K of these mini-slots (i.e., $T = K\beta$), i.e. the channel's coherence time is on the order of K mini-slots. Each splitting algorithm specifies a range of channel gains $[h_i, h_i]$ for each user i, during each mini-slot. The user will only transmit in the mini-slot if its channel gains lie within this range. At the end of a mini-slot, the base station sends $(0, 1, e)$ feedback indicating idleness, a success or a collision, respectively. If a success occurs, the successful user will continue transmitting for the remainder of the time-slot. Otherwise, all of the users update their ranges according to the specific splitting algorithm.

For a homogeneous network, the splitting algorithm in [7] can be viewed as solving (1), when for all i, $\bar{F}_i(h) = \bar{F}(h)$ and $p_i = 1/n$. In that case, $v_1 = v_2 = \cdots = v_n = 0$ and (2) is equivalent to

$$A(\mathbf{H}) = \arg \max_{i=1} \{H_i\},$$

i.e., in this case a centralized scheduler that schedules the user with the best channel gain in each time-slot is distribution fair. In the distributed splitting algorithm in [7], every user has the same range $[h_l, h_h]$; whenever a collision occurs, this range is split into two parts and the users with channel gains in upper part transmit in the next mini-slot. As discussed above, it can be shown that this algorithm will require on average no more

than 2.509 mini-slots to find the user with the best channel gain, independent of the channel distribution or the number of users, n.

In the heterogeneous case, we assume that the users are classified into K different classes according to their service priorities. For $k = 1, \ldots, K$, let $\mathcal{C}_k$ denote the set of class k users, and let $n_k = |\mathcal{C}_k|$ be the number of class k users. Each user i in class k, has the same priority; with some abuse of notation, we indicate this priority by p_k so that $p_i = p_k$ for all $i \in \mathcal{C}_k$. The class with higher priority has a larger p_k in (1), and accordingly, a smaller v_k in (2). We assume that all users know the total number of users in each class, n_k as well as the corresponding parameter v_k. With a small finite number of classes that change slowly, the additional overhead to convey this will be small.

The scheduling rule in (2) can be viewed as finding the minimum of a set of random variables $\{X_j\}$, where for each $j \in \mathcal{C}_k$, $X_j = \bar{F}_j(h_j) + v_k$ is an i.i.d. uniform random variable on $[v_k, 1 + v_k]$. For the heterogeneous case, we describe our splitting algorithms in terms of these $\{X_j\}$ random variables. Specifically, each user will keep track of two thresholds X_L and X_H, and only transmit when $X_j \in [X_L, X_H]$. These thresholds will be the same for each user; however, the corresponding channel thresholds $h_L = \bar{F}_i^{-1}(X_L - v_k)$ and $h_H = \bar{F}_i^{-1}(X_H - v_k)$ may now be different for different users.

A. Known number of users per collision

We first give a splitting algorithm for the idealized case where the number of users in each class involved in each collision is known, i.e. this information is somehow measured and feedback to the users after each mini-slot. Subsequently, we will relax this assumption and give an algorithm for the case where this information is not known. To simplify our discussion, we consider a case with only $K = 2$ classes, with $v_1 = 0$ and $v_2 = v$. At the end of this section we comment briefly on how to extend these results to an arbitrary number of classes.

At the beginning of each time-slot for $i \in \mathcal{C}_1$, X_i is uniformly distributed on $[0, 1]$ and for $i \in \mathcal{C}_2$, X_i is uniformly distributed on $[v, 1+v]$. Assume at a given mini-slot, we know there are m_1 class one users whose values X_i lie in the interval $[a_1, b_1] \subseteq [0, 1]$ and m_2 class two users with X_i values in the interval $[a_2, b_2] \subseteq [v, 1 + v]$, where $a_1 \le a_2 \le b_1 \le b_2$. For $i = 1, 2$, let $F_i(x|a_i, b_i)$ be the conditional c.d.f. of X_i for the class i users, conditioned on X_i being in the interval $[a_i, b_i]$, i.e., $F_i(x|a_i, b_i)$ will be the uniform c.d.f. on $[a_i, b_i]$, so that

$$F_i(x|a_i, b_i) = \frac{x - a_i}{b_i - a_i}.$$

For $x \in [a_1, b_2]$, let

$$
\hat{F}_{1,2}(x|a_1, a_2, b_1, b_2) \\
= \frac{m_1}{m_1 + m_2} F_1(x|a_1, b_1) + \frac{m_2}{m_1 + m_2} F_2(x|a_2, b_2), \quad (3)
$$

so that $\hat{F}_{1,2}(x|a_1, a_2, b_1, b_2)$ is the average of the conditional c.d.f.s of all the users in the indicated intervals.

We develop our splitting algorithm by considering a homogeneous system with $m_1 + m_2$ users with X values in the range $[a_1, b_2]$ and the c.d.f. $\hat{F}_{1,2}(x|a_1, a_2, b_1, b_2)$. In this homogeneous system, the probability of a success in the next time-slot is maximized by allowing only the users in the interval $[a_1, b^*]$ to transmit, where b^* is is chosen so that each user transmits with probability $\frac{1}{1+m_2}$. This is given by the following "splitting" function:

$$
b^* = \mathrm{split}(a_1, a_2, b_1, b_2, m_1, m_2) \\
= \hat{F}_{1,2}^{-1}\left(\frac{1}{m_1 + m_2}\Big|a_1, a_2, b_1, b_2\right).
$$

Using this function our first splitting algorithm is given in Algorithm 1. Here $a \vee b = \max(a, b)$. Also, note that $X_L \le X_P \le X_H$, where X_P indicates the highest value for which it is known that at least one user has $X_j < X_P$, due to a previous collision.

Algorithm 1 Generalized Splitting Algorithm I:

> *initialize:* $X_L = 0$, $X_H = 1$ and $X_P = \mathrm{split}(0, v, 1, 1 + v, n_1, n_2)$
> **while** $m \ne 1$ and $k \le K$ **do**
>   **if** $X_L \le X_j \le X_P$ **then**
>   Transmit
>   **end if**
>   Receive feedback $m \in (0, 1, e)$, m_1 and m_2;
>   **if** $m = e$ **then**
> $X_H = X_P$, $X_P = \mathrm{split}(X_L, X_L \vee v, X_P, X_H, m_1, m_2)$;
>   **else if** $m = 0$ **then**
>   **if** $X_H \ne 1$ **then**
> $X' = \mathrm{split}(X_P, X_P \vee v, X_H, X_H, m_1, m_2)$;
>   **else**
> $X' = \mathrm{split}(X_P, X_P \vee v, 1, 1 + v, n_1, n_2)$;
>   **end if**
> $X_L = X_P$;
> $X_P = X'$;
>   **end if**
>   k=k+1
> **end while**
> **if** Transmitted in the last mini-slot and $m = 1$ **then**
>   Continue transmitting until $k = K$
> **end if**

In Algorithm 1, all users with X_j values less than $X_P = \mathrm{split}(0, v, 1, 1+v, n_1, n_2)$ will transmit in the first mini-slot. If more than one user transmits ($m = e$), then the interval $[0, X_P]$ will be split and users in the lower portion will transmit in the next mini-slot. If no users transmit, then the interval $[X_P, 1]$ will be split. The algorithm will continue in this fashion, until either there are no more mini-slots in the current time-slot or a success occurs. Note that if, for example, after the first mini-slot, a collision occurs then we know that the m_1 class 1 users have X_j values in the range in $[0, X_P]$, while the m_2 class 2 users have X_j values in the range $[v, X_P]$. This was the reason for including the a_i and b_i parameters in the definition of the

split function. It is easy to see that when a success occurs it must be the same user selected by the scheduling rule in (2). Next we analyze the average number of mini-slots needed to find this user.

Let $E(v, n_1, n_2)$ denote the average number of mini-slots required by Algorithm 1 for a system with n_1 class one users, n_2 class two users, an offset of v and an infinite number of mini-slots/time-slot. Note that when there are a finite number of mini-slots/time-slot, the algorithm will require fewer mini-slots due to those times that it stops because $k = K$. When $v = 0$, this splitting algorithm can be shown to be equivalent to the basic algorithm from [7] applied to a homogeneous system with $n_1 + n_2$ users, when the number of users involved in a collision is known.[1] We will prove that $E(v, n_1, n_2) \leq E(0, n_1, n_2)$ for any $v \in [0, 1]$. This result implies that the average number of mini-slots used by Algorithm 1 is no greater than in the homogeneous case, and so is also bounded by 2.509.

Our proof uses the following theorem for independent, non-identically distributed random variables from [12]. Suppose that $X_1, X_2, ..., X$ are independent random variables with corresponding continuous c.d.f.s $F(x)$, for $i = 1, .., n$. Let $\mathbf{F} = \{F(x)\}_{=1}$ and let $\hat{F}(x) = \frac{1}{}\sum_{=1} F(x)$ denote the average c.d.f. of these random variables. We use $\Pr\{X \in A|\mathbf{F}\}$ to denote the probability that $X \in A$ when $\{X\}_{=1}$ have the c.d.f.'s in $\mathbf{F}$ and $\Pr\{X \in A|\hat{F}\}$ to denote this probability when $\{X\}_{=1}$ all have the c.d.f. $\hat{F}$. Let $X_{(1)} \leq X_{(2)}... \leq X_{(\ )}$ denote the n order statistics corresponding to $\{X\}_{=1}$. The following result relates the distribution of these order statistics under $\mathbf{F}$ with that under $\hat{F}$.

Theorem 1 ([12]): For $r = 2, 3, ..., n - 1$ and all $x \leq \hat{F}^{-1}(\frac{-1}{}) < \hat{F}^{-1}(\frac{-}{}) \leq y$,

$$\Pr\{x < X_{(\ )} \leq y|\mathbf{F}\} \geq \Pr\{x < X_{(\ )} \leq y|\hat{F}\},$$

where equality holds only if $F_1(u) = F_2(u) = ... = F(u)$ at both $u = x$ and $u = y$. Also, for all x,

$$\Pr\{X_{(1)} \leq x|\mathbf{F}\} \geq \Pr\{X_{(1)} \leq x|\hat{F}\},$$

and

$$\Pr\{X_{(\ )} \leq x|\mathbf{F}\} \leq \Pr\{X_{(\ )} \leq x|\hat{F}\},$$

with strict inequalities unless $F_1(x) = F_2(x) = ... = F(x)$ at x.

Let $x = \hat{F}^{-1}(\frac{1}{})$,

$$P = \Pr\{\text{exactly } m \ X\text{'s satisfy } X < x|\mathbf{F}\},$$

and let,

$$P' = \Pr\{\text{exactly } m \ X\text{'s satisfy } X < x|\hat{F}\}.$$

Using these quantities, we then have the following corollary:

Fig. 2. The average number of mini-slots required for Algorithm 1 for a one-class (homogeneous) system and a two class (heterogeneous) system as a function of the total number of users in the system.

Corollary 1: For $r = 2, 3, .., n - 1$, $\sum_{=} P \leq \sum_{=} P'$, $P_0 \leq P'_0$, and $P \leq P'$.

The proof is given in Appendix I.

In terms of the splitting algorithm, one can view P as the probability that m users in a heterogeneous network will transmit when the splitting range is $[0, x]$, and P' is the corresponding probability for a homogeneous network. With the aid of this corollary, we prove the following proposition:

Proposition 1: For $0 \leq v \leq 1$, and all n_1 and n_2,

$$E(v, n_1, n_2) \leq E(0, n_1, n_2).$$

The proof is given in Appendix II.

A simulation result illustrating this proposition is shown in Figure 2. This shows $E(0.1, \frac{}{2}, \frac{}{2})$ and $E(0, \frac{}{2}, \frac{}{2})$ as a function of the total number of users in the system, n. It can be seen that the average number of mini-slots required in the heterogeneous case is slightly less than in the homogeneous case, and both are less than the bound of 2.509. The difference between the homogeneous case and the heterogeneous case appears to be fairly constant as the number of users in the system varies.

Though we presented this algorithm for the case where there are $K = 2$ classes of users, the algorithm and our analysis can easily be extended to an arbitrary number of classes. We briefly sketch how this can be done. Consider a system with K classes and, without loss of generality, assume that $p_1 \geq p_2 \geq \cdots \geq p_{\ c}$, so that the parameters $\{v\}$ satisfy $0 = v_1 \leq v_2 \leq \cdots \leq v_{\ c}$. In this case, at a given mini-slot, based on the feedback, we will know that for each class i, there are m users with X values in the interval $[a, b]$, where

$$a_1 \leq a_2 \cdots \leq a_{\ c} \leq b_1 \leq b_2 \cdots \leq b_{\ c}.$$

Each of the m class i users will have a uniform conditional

c.d.f. $F(x|a, b)$ on the interval $[a, b]$. Using these, we define

$$\hat{F}_{\mathbf{m}}(x|\mathbf{a},\mathbf{b}) = \sum_{=1}^{c} \frac{m}{\sum_{=1}^{c} m} F(x|a, b),$$

where $\mathbf{a} = (a_1, \ldots, a_c)$, $\mathbf{b} = (b_1, \ldots, b_c)$ and $\mathbf{m} = (m_1, \ldots, m_c)$. The splitting function for this case is then given by

$$\text{split}(\mathbf{a}, \mathbf{b}, \mathbf{m}) = \hat{F}_{\mathbf{m}}^{-1}\left(\frac{1}{\sum_{=1}^{c} m}\bigg|\mathbf{a}, \mathbf{b}\right).$$

Using this function, Algorithm 1 can be generalized to allow for more than two classes. Note the arguments of the split function will also need to be modified to take into account the different parameters v for each class. Theorem 1 and Corollary I hold for any number of classes. Proposition 1 can be easily generalized to an arbitrary number of classes as well.

B. Number in collision not known

In the previous section, we assumed that after a collision occurs, the number of users of each class involved in the collision is fed back to all the users. This may be hard to realize in practice. Therefore, we next consider the case where the numbers of users involved in a collision (and their respective classes) is unknown. However, we emphasize that the initial number of users in each class is still known. Again this is reasonable if the number of active users changes on a much slower time-scale. It follows that until the first collision occurs, the same procedure as in Algorithm 1 can be used. However, once a collision occurs it is no longer known how many users of each class were involved. We note that when a collision occurs, assuming that each user transmits with a probability less than $\frac{3}{+1}$ (where n is the total number of users), then the highest probability event is that only two users were involved in this collision. Accordingly, we modify the splitting algorithm to use the function $\text{split}_2(X, X)$ defined by

$$\text{split}_2(X, X) = X + (X + X)/2. \tag{4}$$

Suppose that there are two users with channel gains in the interval $[X, X]$ and conditioned on being in this interval, their X values are uniformly distributed. In this case, the function in (4) results in a new interval that maximizes the probability of success in the next time-slot. However, note that with fairness constraints, depending on the class of the users involved in a collision, each user may not have a X value that is uniformly distributed on the entire interval. The resulting splitting algorithm is given in Algorithm 2. Again, we state this algorithm for the case where there are only $K = 2$ classes; using similar ideas as in the previous section this can be directly extended to more than two classes.

Until the first collision, this algorithm follows the same procedure as Algorithm 1. Thus, each user will transmit if their X value is less than

$$x = \check{F}_{1\,2}^{-1}\left(\frac{1}{n_1, n_2}\bigg|X, X \vee v, 1, 1 + v\right).$$

Algorithm 2 Generalized Splitting Algorithm II:

> *initialize* $X = 0$, $X = 1$, $X = \text{split}(0, v, 1, 1 + v, n_1, n_2)$;
> **while** $m \neq 1$ and $k \leq K$ **do**
>   **if** $X \leq X \leq X$ **then**
>   Transmit;
>   **end if**
>   Receive feedback $m = (0, 1, e)$;
>   **if** $m = e$ **then**
> $X = X$, $X = \text{split}_2(X, X)$;
>   **else if** $m = 0$ **then**
>   **if** $X \neq 1$ **then**
> $X' = \text{split}_2(X, X)$;
>   **else**
> $X' = \text{split}(X, X \vee v, 1, 1 + v, n_1, n_2)$;
>   **end if**
> $X = X$; $X = X'$;
>   **end if**
> $k = k + 1$;
> **end while**
> **if** Transmitted in the last mini-slot and $m = 1$ **then**
>   Continue transmitting until $k = K$.
> **end if**

In this case, Corollary 1 applies so that $P_0 \leq P_0'$ and $P_1 \geq P'1$. Therefore, we have:

Lemma 1: For Algorithm 2, the average number of mini-slots until a non-idle response is received in a heterogeneous system is smaller than in a homogeneous system with the same number of users. Likewise, the probability that the first non-idle response is a success is larger in a heterogeneous system than in a homogeneous system with the same number of users.

For a single class of users, this algorithm also reduces to one given in [7], where the number of user involved in a collision is not known. In that case, it was shown that without knowledge of the number involved in a collision, the average number of mini-slots is still upper-bounded by 2.509 for any number of users and fading distribution. Next we present some simulation results of this algorithm for both a one class and two class system with the same total number of users. These show that, with multiple classes, the average number of mini-slots required can be smaller or larger than in the corresponding homogeneous case. In these simulations, the lower priority users have an offset of $v = 0.1$. As shown in Figure 3, with an equal number of users in both classes, the average number of mini-slots required is again generally smaller than in the homogeneous case. Here, in both cases, the average number is less the bound of 2.509 for the homogeneous case.

In Figure 4, we fix the number of class one users to be $n_1 = 5$, and vary the number of class two users. In this case, the average number of mini-slots required is larger than in the homogeneous case with $n_1 + n_2$ users, and with sufficiently many users it exceeds the bound of 2.509. However, it can be seen that still only a small number (less than 2.7) of mini-slots are required for both cases. From Lemma 1, we see that until
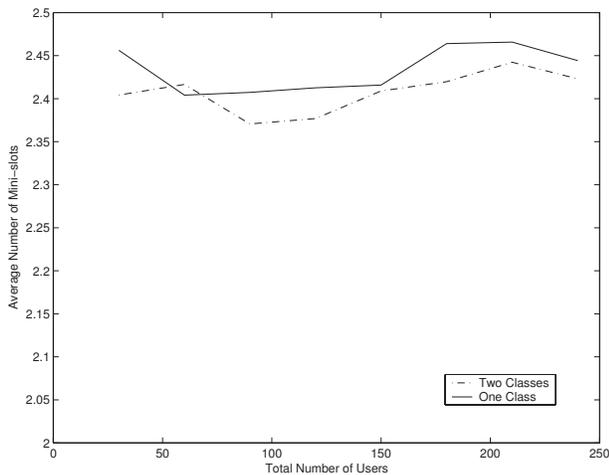
Fig. 3. Average number of mini-slots required by Algorithm 2 as a function of the total number of users with an equal number of users for each class.
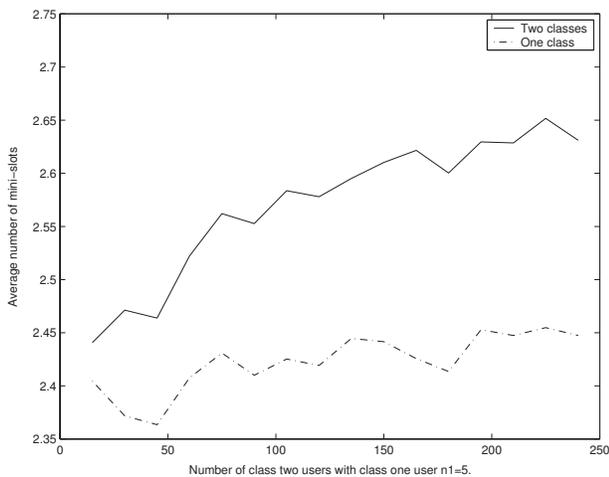
Fig. 4. Number of mini-slots required by Algorithm 2 as a function of the number of users in class two with the number of class one users equal to 5.

the first collision occurs, the algorithm for the two class system is performing on average no worse than in a one class system. However, once a collision occurs, the algorithm may perform worse than in the homogeneous case. The reason for this is that the new splitting function in (4) is based on the assumption that the users involved in a collision all have channel gains uniformly distributed on $[X, X]$. As noted above, due to fairness constraints this may not be the case. In particular, for this two class example whenever there is a class two user in this interval and $X_1 < v$, the class 2 user will transmit with a probability smaller than $1/2$ in the next mini-slot. In Figure 4, as the number of class 2 users increases, the likelihood that one or more users in a collision is class 2 increases, and we see that the average number of mini-slots also increases. Nevertheless, when a collision happens, with high probability there will still be only a small number of users involved in the collision. So the average number of mini-slots required in total is still small.

One possibility to improve on the performance in a case such as that shown in Figure 4 is to use a splitting function that takes into account the initial distribution of users in each class. For example, one possibility would be to replace $\text{split}_2(X, X)$ with

$$\widehat{\text{split}}_2(X, X) = \hat{F}^{-1}_{1\ 2}\left(\frac{1}{2}\bigg|X, X \vee v, X, X\right),$$

where $\hat{F}_{1\ 2}(\cdot)$ is the average c.d.f. of all the users as defined in (3). Characterizing the performance of such an algorithm is a topic of current work. Finally we note that this modification can also be directly extended to more than two classes using the same approach as in the previous section.

IV. CONCLUSIONS

In this paper, we considered opportunistic splitting algorithms for distributively scheduling users in a heterogeneous wireless network, where users have independent but non-identical channel statistics. Our algorithms also take into account temporal fairness constraints across the users. For a system where the number of users in each fairness class involved in each collision is known, we given a splitting algorithm that performs better in a heterogeneous system than in a corresponding homogeneous one, and requires on average no more than 2.509 mini-slots to schedule the right user. When the number of users involved in a collision is not known, we give an algorithm that may perform worse in a heterogeneous environment, but still appears to require only a small amount of overhead on average.

Here we focused on a back-logged model and assumed that the fading was slow enough for many mini-slots/coherence time. Relaxing these assumptions are possible directions for future work.

APPENDIX I
PROOF OF THE COROLLARY 1

Proof: For $r = 2, 3, ..., n-1$, $x = \hat{F}^{-1}(\frac{1}{}) \leq \hat{F}^{-1}(\frac{-1}{})$, and so, using Theorem 1 with $y = \infty$, we have

$$\Pr\{X_{(\ )} > x|\mathbf{F}\} \geq \Pr\{X_{(\ )} > x|\hat{F}\}.$$

Therefore,

$$\Pr\{X_{(\ )} \leq x|\mathbf{F}\} \leq \Pr\{X_{(\ )} \leq x|\hat{F}\}.$$

If $X_{(\ )} < x$, then at least r users must have $X < x$. Hence,

$$Pr\{X_{(\ )} < x|\mathbf{F}\} = \sum_{=} P,$$

and

$$Pr\{X_{(\ )} < x|\hat{F}\} = \sum_{=} P'.$$

Therefore, for $r = 2, 3, .., n-1$, we have

$$\sum_{=} P \leq \sum_{=} P'.$$

Likewise, from Theorem 1,

$$\Pr\{X_{(1)} \le x | \mathbf{F}\} \ge \Pr\{X_{(1)} \le x | \hat{F}\},$$

and so

$$\sum_{=1} P \ge \sum_{=1} P'.$$

Therefore $P_0 \le P_0'$.

Finally, for $r = n$, Theorem 1 states that

$$\Pr\{X_{(\ )} \le x | \mathbf{F}\} \ge \Pr\{X_{(\ )} \le x | \hat{F}\},$$

and so $P \le P'$. ∎

APPENDIX II
PROOF OF THE PROPOSITION 1

Proof: In each iteration of the splitting algorithm, when we call the function $\mathrm{split}(a_1, b_1, a_2, b_2, m_1, m_2)$, there are m_1 class one users in range $[a_1, b_1]$ and m_2 class two users in range $[a_2, b_2]$. Without loss of generality, we can normalize the range $[a_1, b_1]$ to be $[0, 1]$ and define two offsets v_1 and v_2, where $v_1 = \frac{2^{-1}}{1^{-1}}$ and $v_2 = \frac{2^{-1}}{1^{-1}}$ so that the class two users have a normalized range of $[v_1, 1 + v_2]$. The average number of mini-slots, given that the users have these normalized ranges, will be the same as with their original ranges. Given this normalization, let $E(v_1, v_2, m_1, m_2)$ denote the average number of mini-slots required for the algorithm to find the best user starting with m_1 users in range $[0, 1]$, and m_2 users in $[v_1, 1 + v_2]$. Note that $E(v, n_1, n_2)$ in the statement of the theorem is equivalent to $E(v, v, n_1, n_2)$ using this notation; i.e. initially $v_1 = v_2 = v$; the subsequent choice of these will depend on the algorithm's progression.

Let

$$E\ (m_1, m_2) = \max_{0 \le\ _1 \le 1\ 0 \le\ _2 \le 1} E(v_1, v_2, m_1, m_2),$$

i.e., $E\ (m_1, m_2)$ is the maximum average number of mini-slots needed by the splitting algorithm over all choices of v_1 and v_2 starting with the given number of users in each class. Also, let $E(0, n)$ be the average number of mini-slots needed for a homogeneous system starting with n users. To prove this proposition, we prove the slightly stronger statement that for all n_1, n_2,

$$E\ (n_1, n_2) \le E(0, n).$$

The desired results then directly follows from the definition of these quantities. We prove this by induction. First, we prove it is true for $n_1 = 1$ and $n_2 = 1$ (Note it is trivially true for $(n_1 = 1, n_2 = 0)$ and $(n_1 = 0, n_2 = 1)$). In this case, after one iteration of the splitting algorithm, either a success occurs or both users are above or below the threshold $X\ = x = \hat{F}_{1\,1}^{-1}(\frac{1}{2})$. Hence,

$$E\ (1,1) \le 1 + E\ (1,1)P_2 + E\ (1,1)P_0,$$

where P is the probability defined before Corollary 1 (For some cases, it may be that $P_2 = 0$). Simplifying we have,

$$E\ (1,1) \le \frac{1}{1 - P_2 - P_0}.$$

Using the same argument for the homogeneous case,

$$E(0, 2) = 1 + P_0' E(0, 2) + P_2' E(0, 2),$$

and so

$$E(0, 2) = \frac{1}{1 - P_0' - P_2'}.$$

Hence, using Corollary 1, $E\ (1, 1) \le E(0, 2)$.

For the induction step, now assume for all $k_1 \le n_1$ and $k_2 \le n_2 - 1$ that $E\ (k_1, k_2) \le E(0, k_1 + k_2)$; we then prove that this must hold for $E\ (n_1, n_2)$ (note the same argument applies if we instead increase the number of class 1 users).[2]

Let

$$E\ (k) = \max\{E\ (k_1, k_2) | k_1 + k_2 = k\}.$$

It follows from the induction hypothesis that

$$E\ (k) \le E(0, k),$$

for all $k < n := n_1 + n_2$.

Again considering all possibilities after one iteration, we have

$$E\ (n_1, n_2) \le 1 + \sum_{=2}^{-1} E\ (i)P + E\ (n_1, n_2)(P + P_0)$$

$$\le 1 + \sum_{=2}^{-1} E(0, i)P + E\ (n_1, n_2)(P + P_0).$$

Thus,

$$E\ (n_1, n_2) \le \frac{1 + \sum_{=2}^{-1} E(0, i)P}{1 - P - P_0}. \tag{5}$$

Likewise, for a homogeneous system with $n = n_1 + n_2$ users,

$$E(0, n) = \frac{1 + \sum_{=2} E(0, i)P'}{1 - P_0'}. \tag{6}$$

From Corollary 1, $\sum_{=2} P' \ge \sum_{=2} P$. Next, we use this to show that

$$\sum_{=2} E(0, i)P' \ge \sum_{=2} E(0, i)P.$$

To do this, we create new p.m.f.'s $\{\tilde{P}'\}_{=0}$ and $\{\tilde{P}\}_{=0}$ so that for $i = 2, \ldots, n$, $\tilde{P}' = P'$ and $\tilde{P} = P$ and so that $\{\tilde{P}'\}$ is stochastically larger than $\{\tilde{P}\}$. Note this stochastic ordering is not true for the original p.m.f.'s since we only have $\sum_{=} P' < \sum_{=} P$, for $r = 2, 3, ..., n$, but we can easily find new choices of $\tilde{P}_0$, $\tilde{P}_0'$, $\tilde{P}_1$, and $\tilde{P}_1'$ so that this is true.

Next, consider the function $f(i)$, given by:

$$f(i) = \begin{cases} 0, & i = 0, 1, \\ E(0, i), & i = 2, \ldots, n. \end{cases}$$

This will be a non-decreasing function of i, and so we can apply the following stochastic ordering result [13]: for two random variables, X and Y, if $\bar{F}\ (a) \ge \bar{F}\ (a)$ for all a,

[2]More precisely, we are doing a double induction on the number of class one users and class two users.

then $E[g(X)] \geq E[g(Y)]$ for all non-decreasing functions g. It follows that

$$\sum_{=0} f(i)\tilde{P}' > \sum_{=0} f(i)\tilde{P} \, .$$

From their definitions, this is equivalent to

$$\sum_{=2} E(0,i)P' \geq \sum_{=2} E(0,i)P \, ,$$

as desired. Using this in (6), along with the fact that $P_0' \geq P_0$, we have

$$E(0,n) \geq \frac{1 + \sum_{=2} E(0,i)P}{1 - P_0} \, .$$

Re-arranging terms, this is equivalent to

$$E(0,n) \geq \frac{1 + \sum_{=2}^{-1} E(0,i)P}{1 - P_0 - P} \, .$$

Combining this with (5), we have $E\;(n_1, n_2) \leq E(0,n)$, which completes the induction step. ∎

References

[1] R. Agrawal, A. Bendeckar, R. La, and V. Subramanian, "A class and channel-condition based weighted proportionally fair scheduler," in: *Teletraffic Engineering in the Internet Era, Proc. of the International Teletraffic Congress – ITC-17*, pp. 553-565, Salvador da Bahia, Brazil, Sept. 2001.

[2] X. Liu, E. Chong, and N. Shroff, "Opportunistic transmission scheduling with resource sharing constraints in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 2053–2064, Oct. 2001.

[3] Y. Liu and E. Knightly, "Opportunistic Fair Scheduling over Multiple Wireless Channels," *Proc. of 2003 IEEE INFOCOM*, pp. 1106-1115, San Francisco, CA, April 2003.

[4] S. Borst, "User level performance of channel-aware scheduling algorithms in wireless data networks," in *Proceeding IEEE INFOCOM'03*, April 2003.

[5] P. Viswanath, D. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Transactions on Information Theory*, vol. 48, Jun. 2002.

[6] X. Qin and R. Berry, "Exploiting multiuser diversity for medium access control in wireless networks," in *Proceeding IEEE INFOCOM*, (San Francisco), Mar. 2003.

[7] X. Qin and R. Berry, "Opportunistic splitting algorithms for wireless networks," in *IEEE INFOCOM 2004*, (Hong Kong, P.R. China), March 2004.

[8] Y. Yu and G. Giannakis, "Opportunistic Access in Slotted Aloha Adapted to Decentralized CSI," *Proc. of IEEE GLOBECOM 2004*, pp. 3332-3336, Nov. 2004.

[9] S. Adireddy and L. Tong, "Exploiting decentralized channel state information for random access," *IEEE Transactions on Information Theory*, vol. 51, pp. 537–561, Feb. 2005.

[10] X. Qin and R. Berry, "Distributed Approaches For Exploiting Multiuser Diversity in Wireless Networks," to appear, *IEEE Transactions on Information Theory*.

[11] X. Qin and R. Berry, "Opportunistic splitting algorithms for wireless networks with heterogeneous users," in *38th Annual Conference on Information Science and Systems(CISS)*, (Princeton University), March 2004.

[12] H. David, *Order Statistics*, pp. 22–23. Wiley, 1981.

[13] S. Ross, *Stochastic Processes, 2nd Ed.*, p. 405. Wiley, 1996.